データマイニングと情報可視化

Week 6

稲垣 紫緒
いながき　しお

理学研究院　物理学部門 / 共創学部
inagaki@phys.kyushu-u.ac.jp
ウェスト1号館 W1-A823号室

授業計画

Week 1: Introduction
Week 2: Basics
データの入手
データ加工処理
可視化による見通し
分析結果の可視化
決定木
相関ルール
クラスタ分析
回帰分析
ニューラルネットワーク
Week 2 & 3
Week 4 & 5
Week 6 & 7
モデル選択
クロスバリデーション
Week 8: 応用

データマイニングの代表的な手法

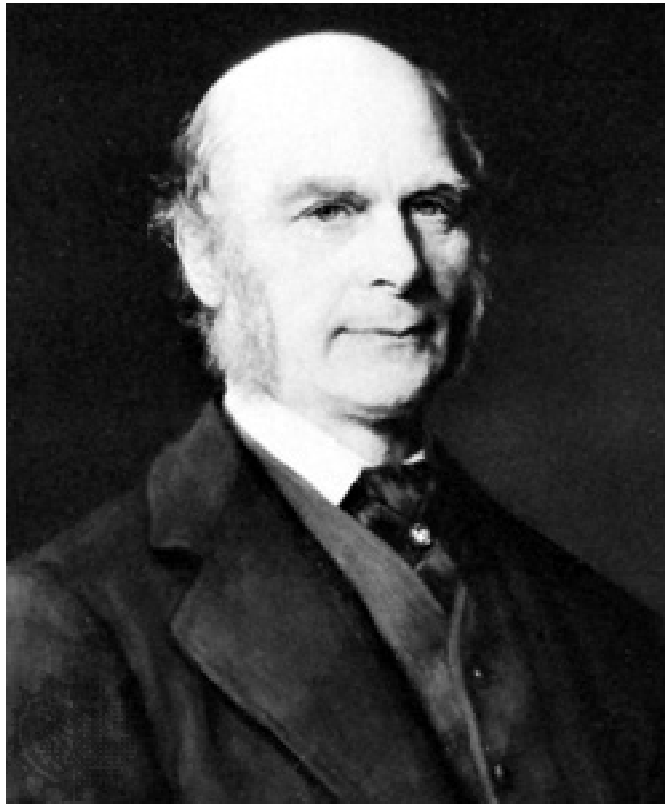
(3) ロジスティック回帰分析

発生確率を予測する

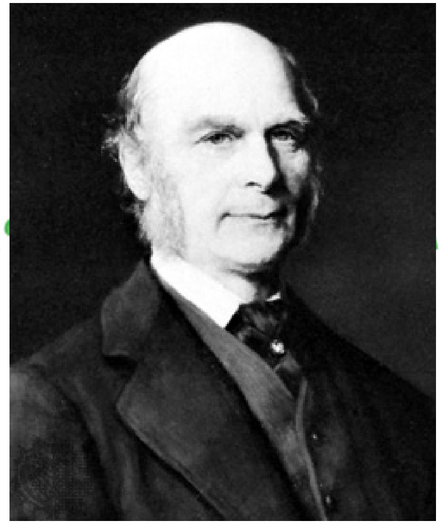
■ がんの発症確率や生存率など
■ アンケート結果から、携帯会社を乗り換える顧客を予測

回帰分析の始まり

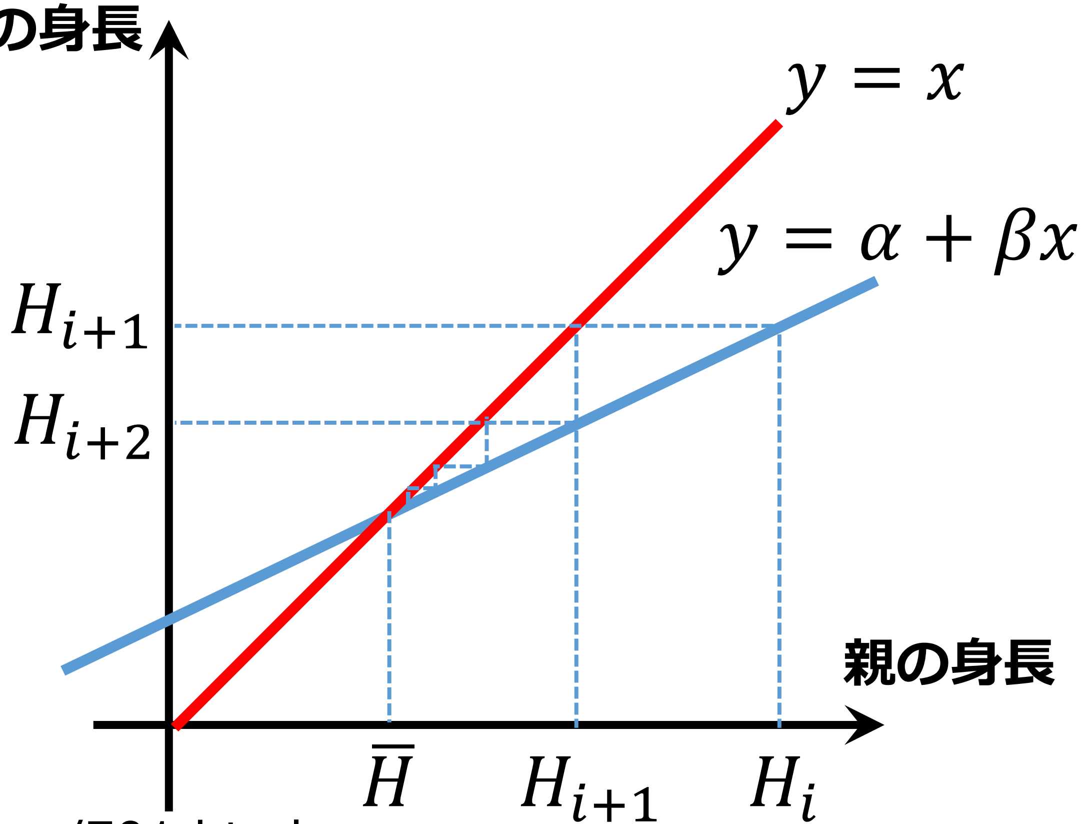
近代統計学の父
フランシス・ゴルドン
Sir Francis Galton
(1822～1911)
チャールズ・ダーウィンの従兄



回帰分析の始まり



- 遺伝研究に統計学的方法を導入
- 「平均回帰」と呼ばれる現象を発見



- 特別に低身長の父親でも、息子たちの身長は父親たちの身長より平均に近くなる→平均回帰「後退（＝regression）」する傾向があることを発見した。
- 相関を表す数値を「相関係数」と名付けた。
- 変数間の解析を分析することを「回帰分析」と呼ぶようになる。

https://www.stat.go.jp/dss/course/701.html

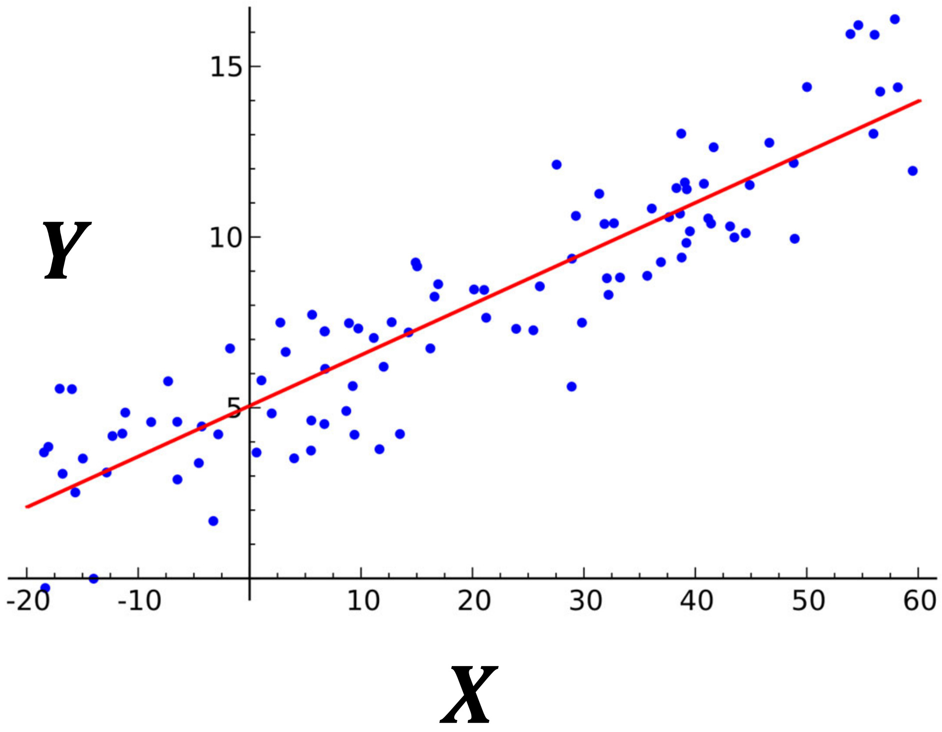
回帰分析による予測

Xの値から、 Yの値を予測したい。

→訓練データから、
$\boldsymbol{\alpha}$と$\boldsymbol{\beta}$を推定する

$$Y = \alpha + \beta X$$

α: 回帰定数
β : 回帰係数



独立変数（説明変数） X
従属変数（目的変数） Y
の間にモデルを当てはめる。

https://ja.wikipedia.org/wiki/%E5%9B%9E%E5%B8%B0%E5%88%86%E6%9E%90

予測とは？

「入力に対する結果を推定する」手法

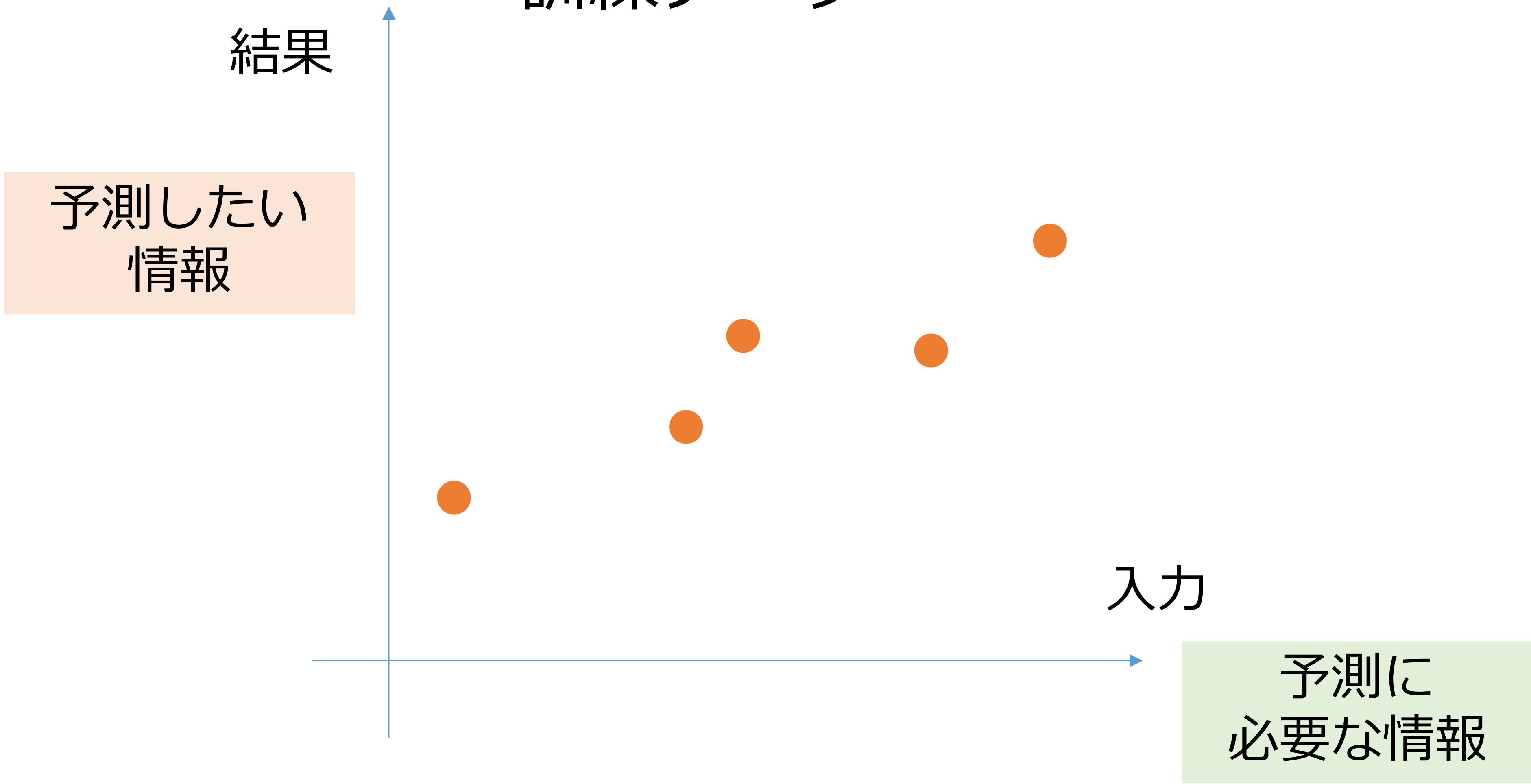
入力
(原因) → 予測器 → 結果

例

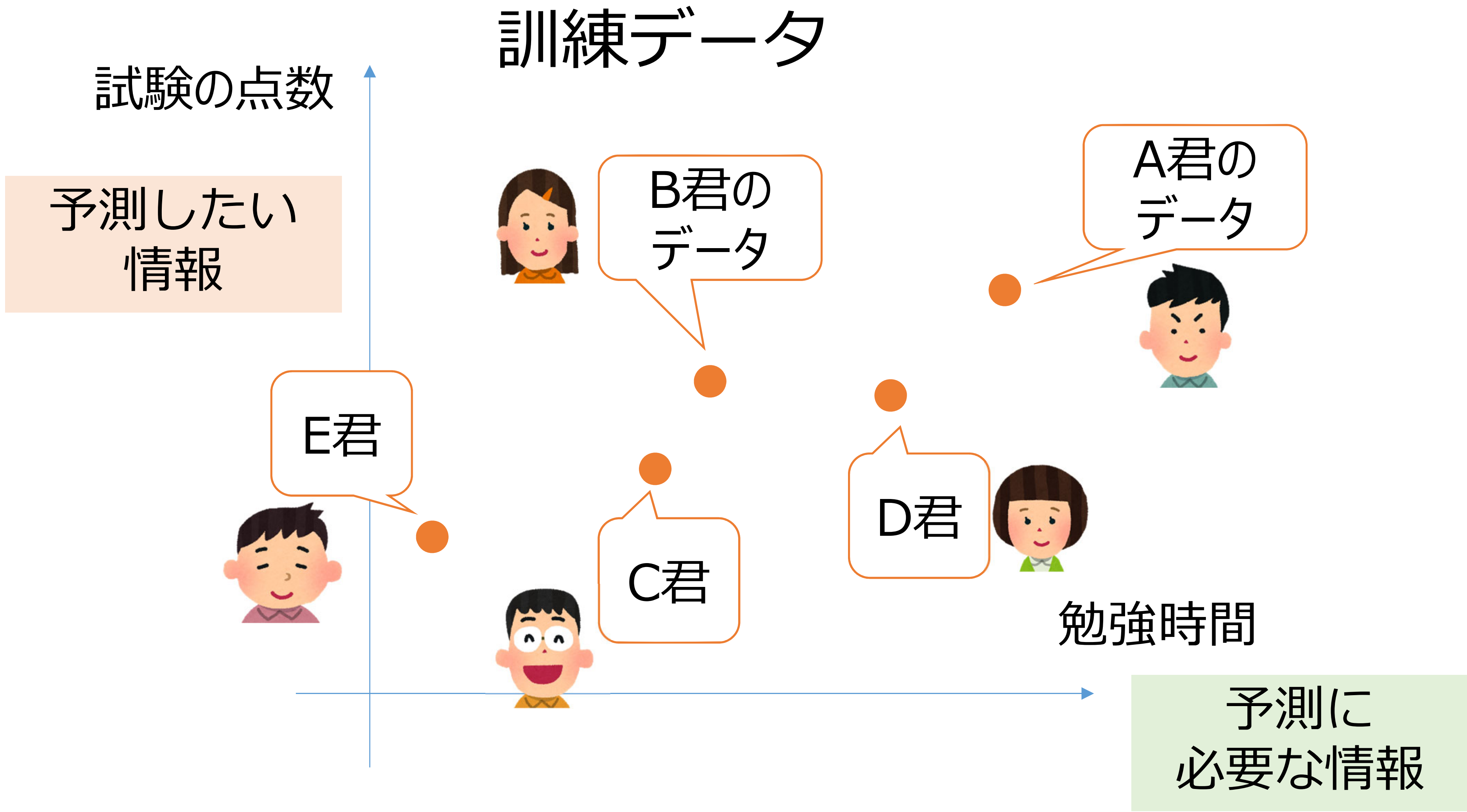
入力(原因)	結果
勉強時間	試験の点数
食事量	糖尿病になる確率
(身長，体重)	バスケットボールの得点
画像	その画像がリンゴの写真である確率

回帰による予測(1/3) データ収集

訓練データ

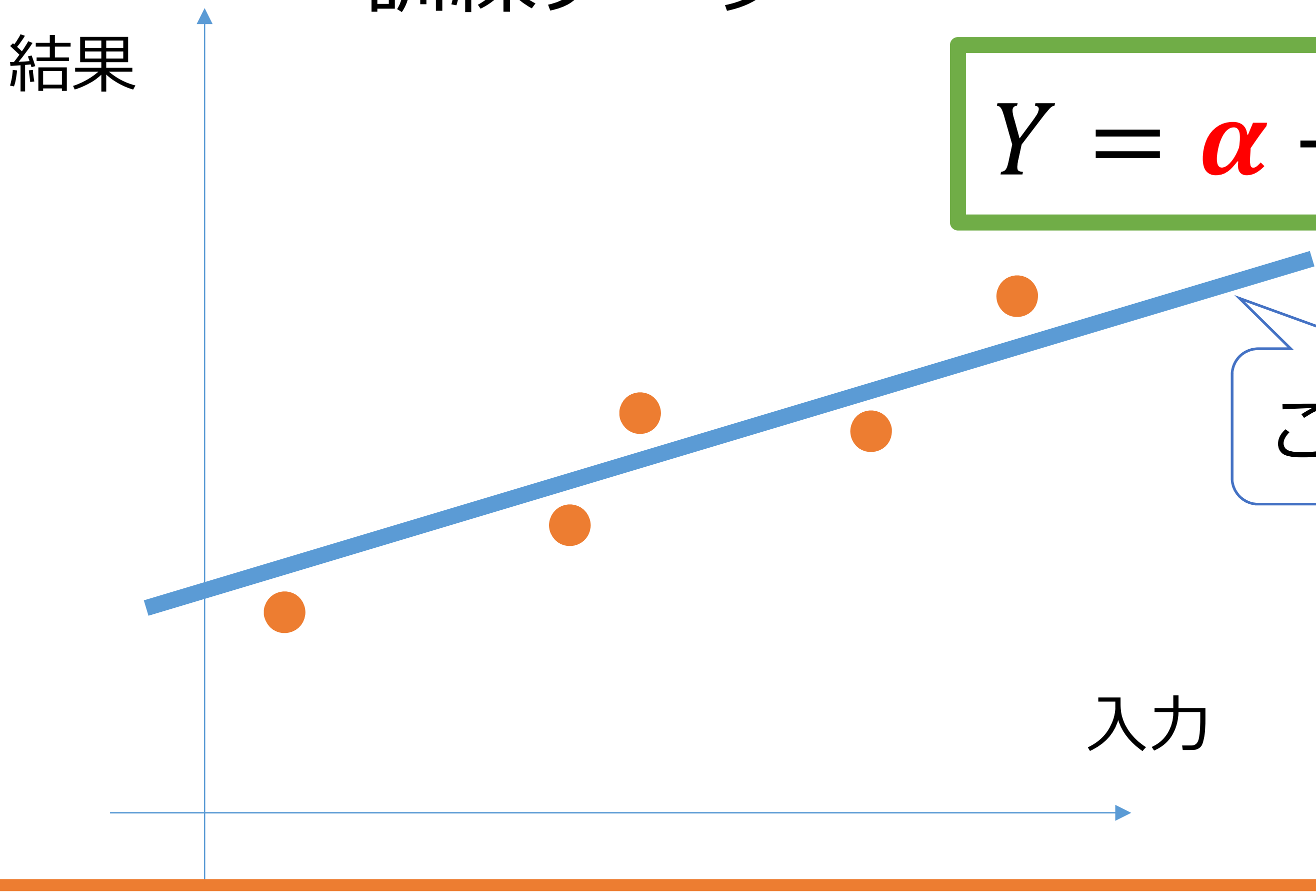


回帰による予測(1/3) データ収集



回帰による予測(2/3) モデル

訓練データ



$$Y = \alpha + \beta X$$

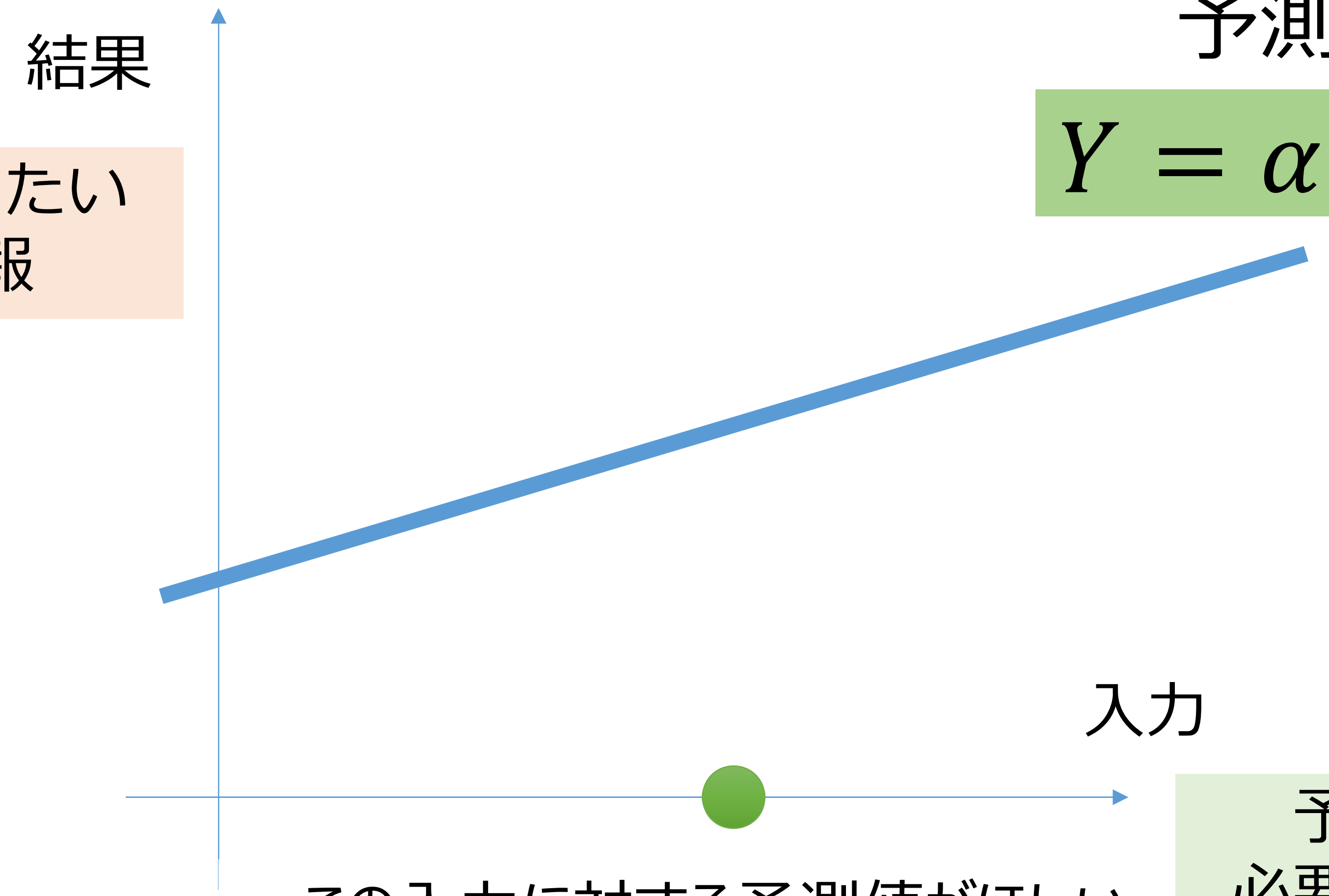
これ大事

モデル＝「条件 or 入力」と「結果」間に成り立つと予想される関係.

上記は「線形モデル」

回帰による予測(3/3) 予測

予測したい
情報



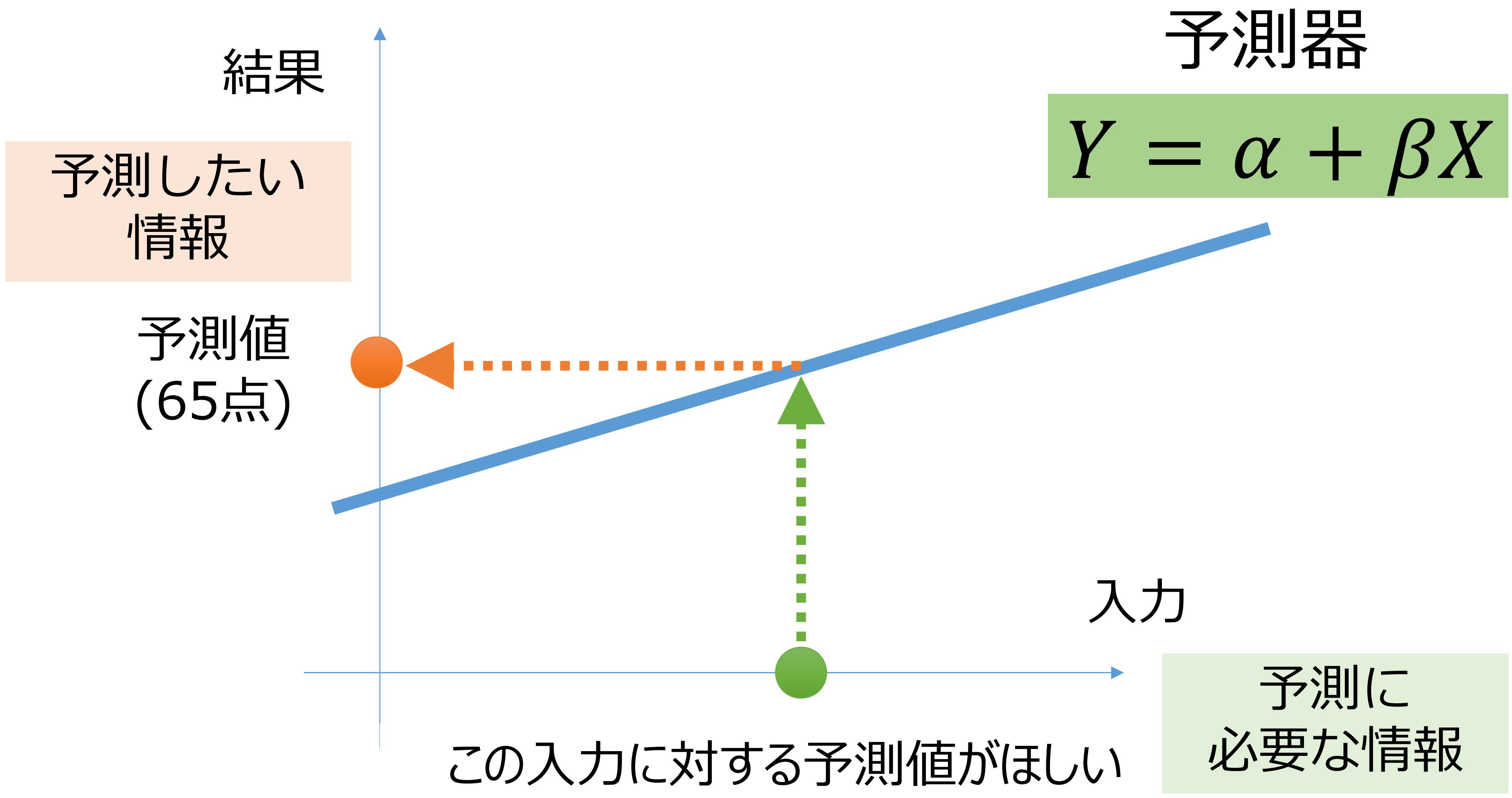
予測器

$$Y = \alpha + \beta X$$

予測に
必要な情報

この入力に対する予測値がほしい
(ex. 勉強3時間だと何点取れそう？)

回帰による予測(3/3) 予測



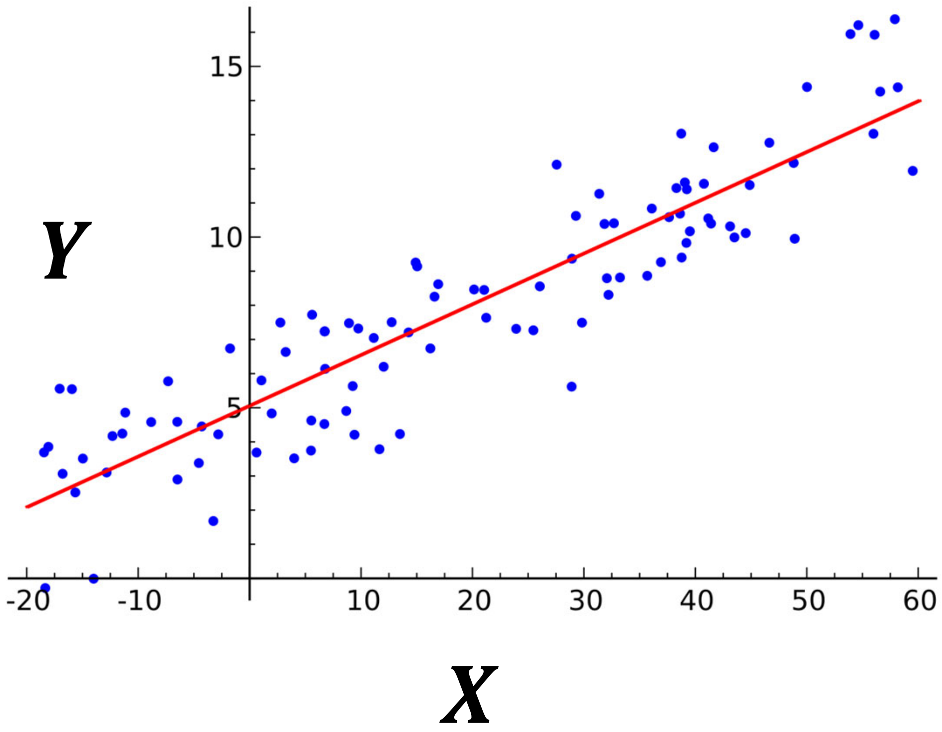
この入力に対する予測値がほしい
(ex. 勉強3時間だと何点取れそう？)

回帰分析による予測

Xの値から、 Yの値を予測したい。
→訓練データから、
$\boldsymbol{\alpha}$と$\boldsymbol{\beta}$を推定する

$$Y = \alpha + \beta X$$

α: 回帰定数
β : 回帰係数



独立変数（説明変数） X
従属変数（目的変数） Y
の間にモデルを当てはめる。

https://ja.wikipedia.org/wiki/%E5%9B%9E%E5%B8%B0%E5%88%86%E6%9E%90

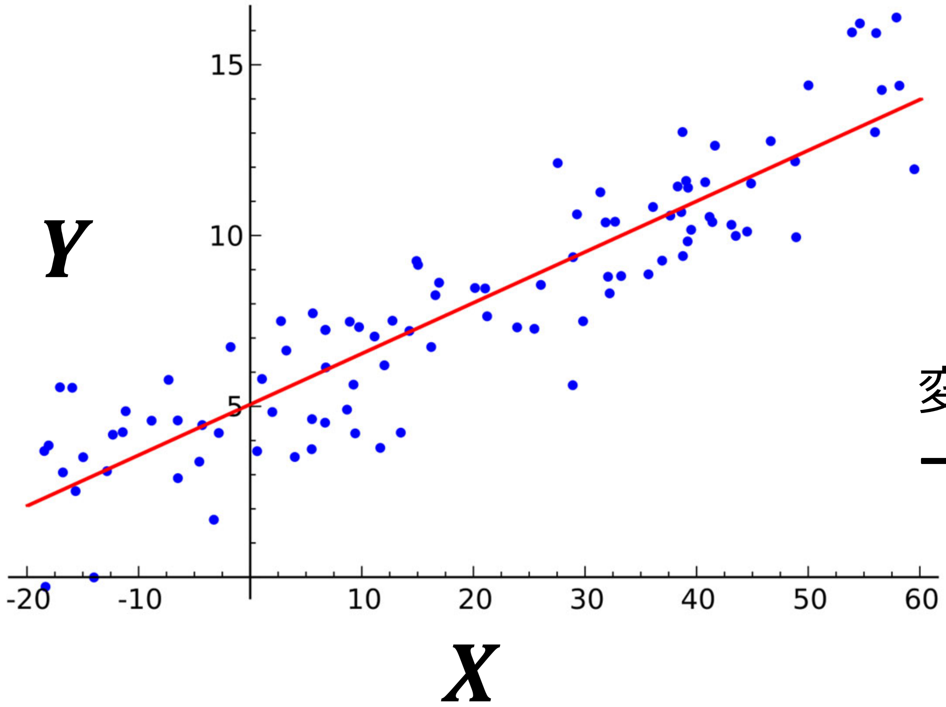
単回帰分析の利用法

天気予報で、来週の予想気温から

- アイスの売り上げを予測する。
  材料をどのくらい用意しておけばいいか??

- ビールの売り上げを予測する
  どのくらいビールを入荷しておいたらいいか??

回帰分析：経済学



消費関数

$Y = \alpha + \beta X$

変数の一次関数で表す
→線形回帰

目的変数：国民所得（ X ）
説明変数：経済全体の消費（ Y ）
消費関数 $Y = \alpha + \beta X$
というモデルで表されるとする。
α, β といった係数を推定する。

回帰分析：線形回帰

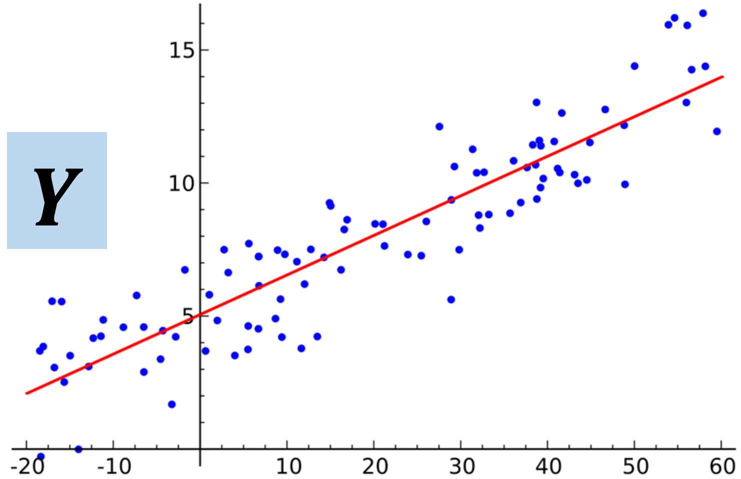
単回帰

説明変数が１つの場合

$$Y = \beta_o + \beta_1 X_1 + \epsilon$$

目的変数 **説明変数**

目的変数 $\boldsymbol{Y}$



$\boldsymbol{X}$ **説明変数**

重回帰

説明変数が複数の場合

$$Y = \beta_o + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \cdots . + \beta_p X_p + \epsilon$$

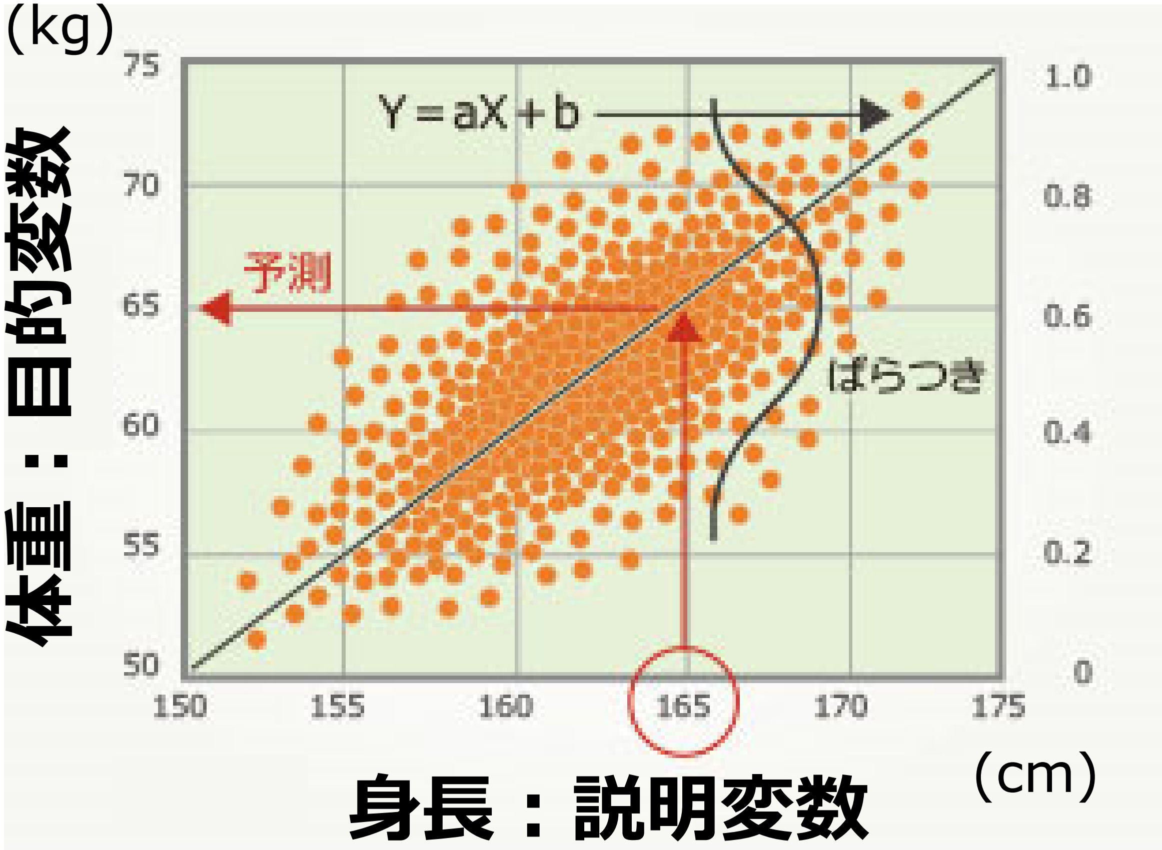
$$Y = \beta_o + \beta_1 X_1 + \beta_1 X_2^2 + \epsilon$$

β_pについて線形(一次)だったら線形回帰

目的変数 **説明変数**

最小二乗法

身長から体重を予測



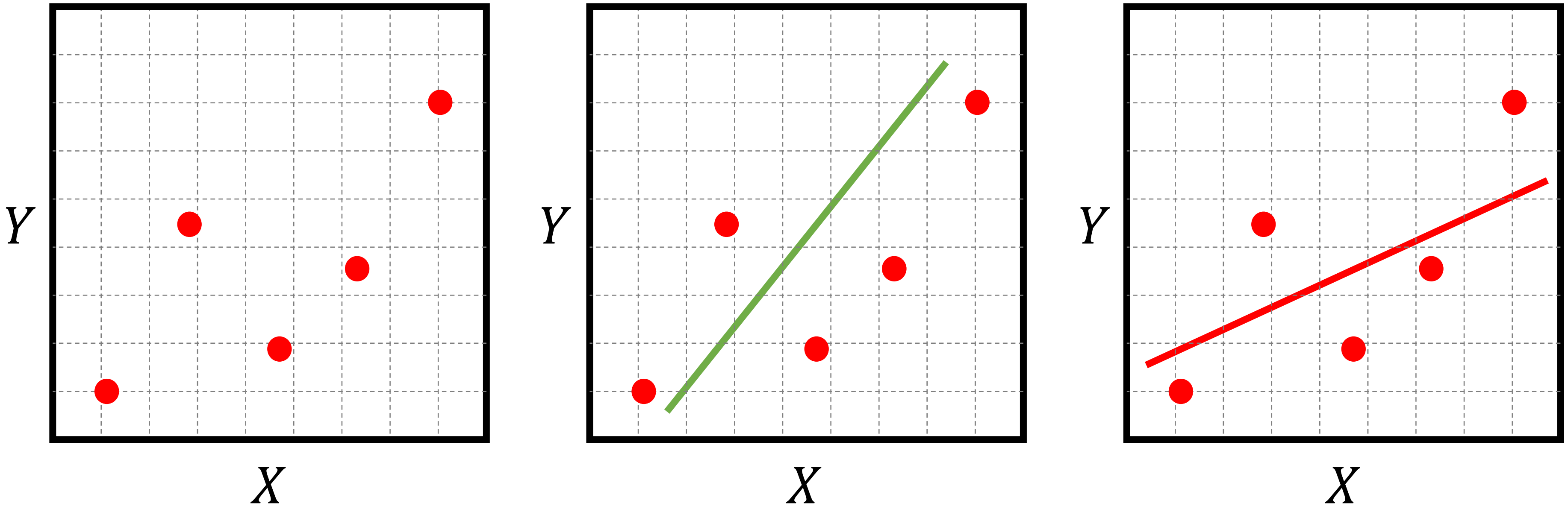
$$Y = aX + b$$

最小二乗法を用いて、
a と b を決定

https://www.albert2005.co.jp/knowledge/statistics_analysis/multivariate_analysis/single_regression

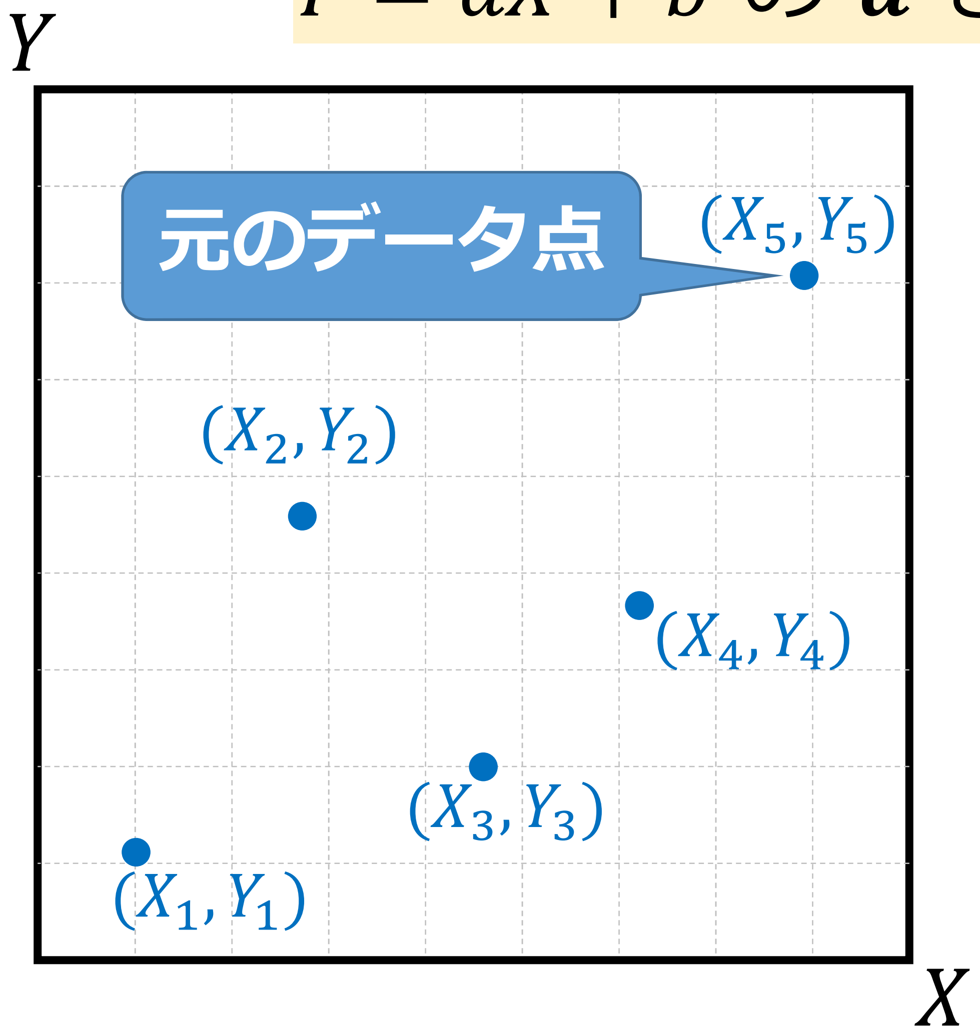
最小二乗法

$Y = aX + b$ の $\boldsymbol{a}$ と $\boldsymbol{b}$ を決めたい。



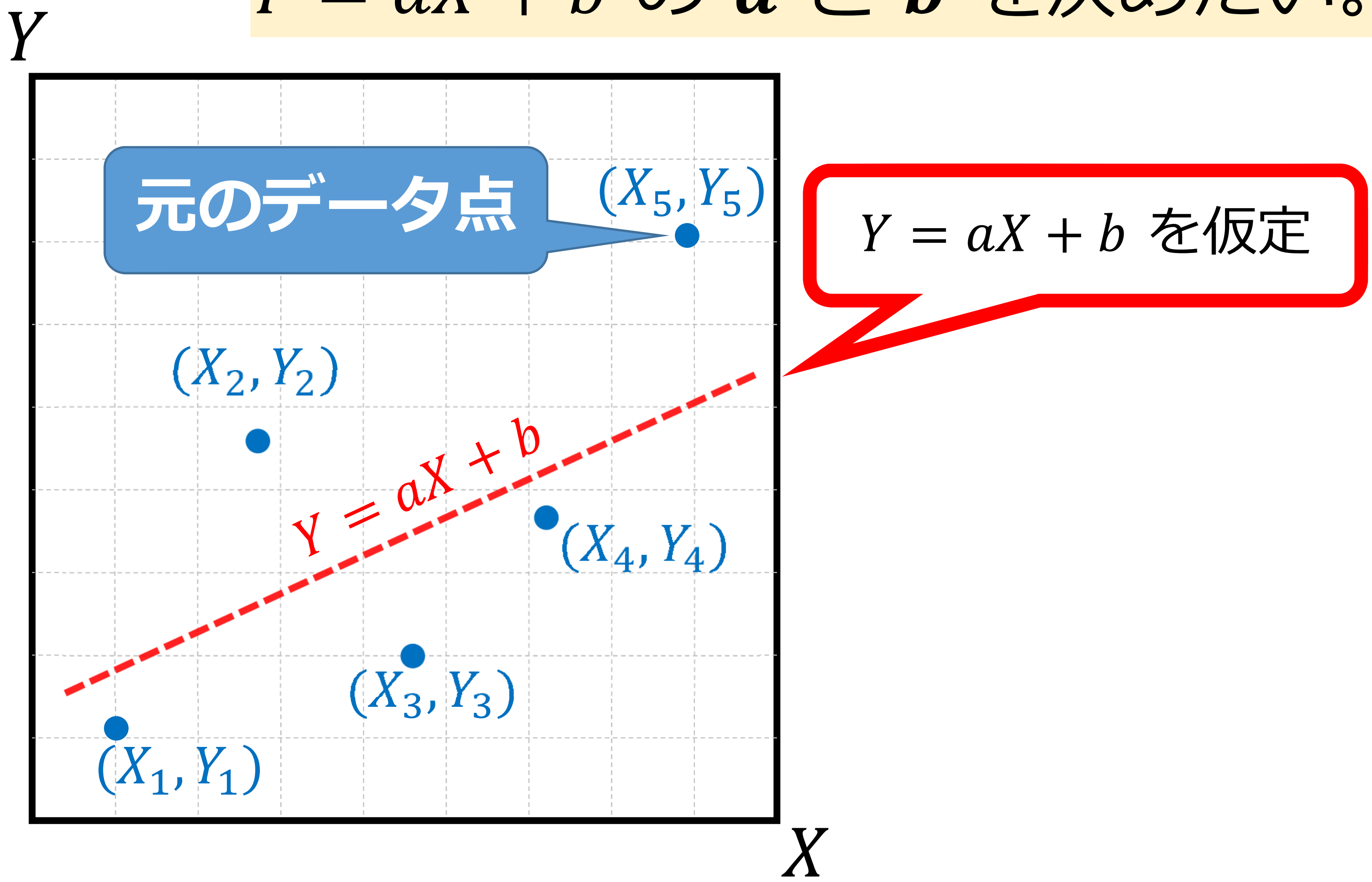
最小二乗法

$Y = aX + b$ の $\boldsymbol{a}$ と $\boldsymbol{b}$ を決めたい。



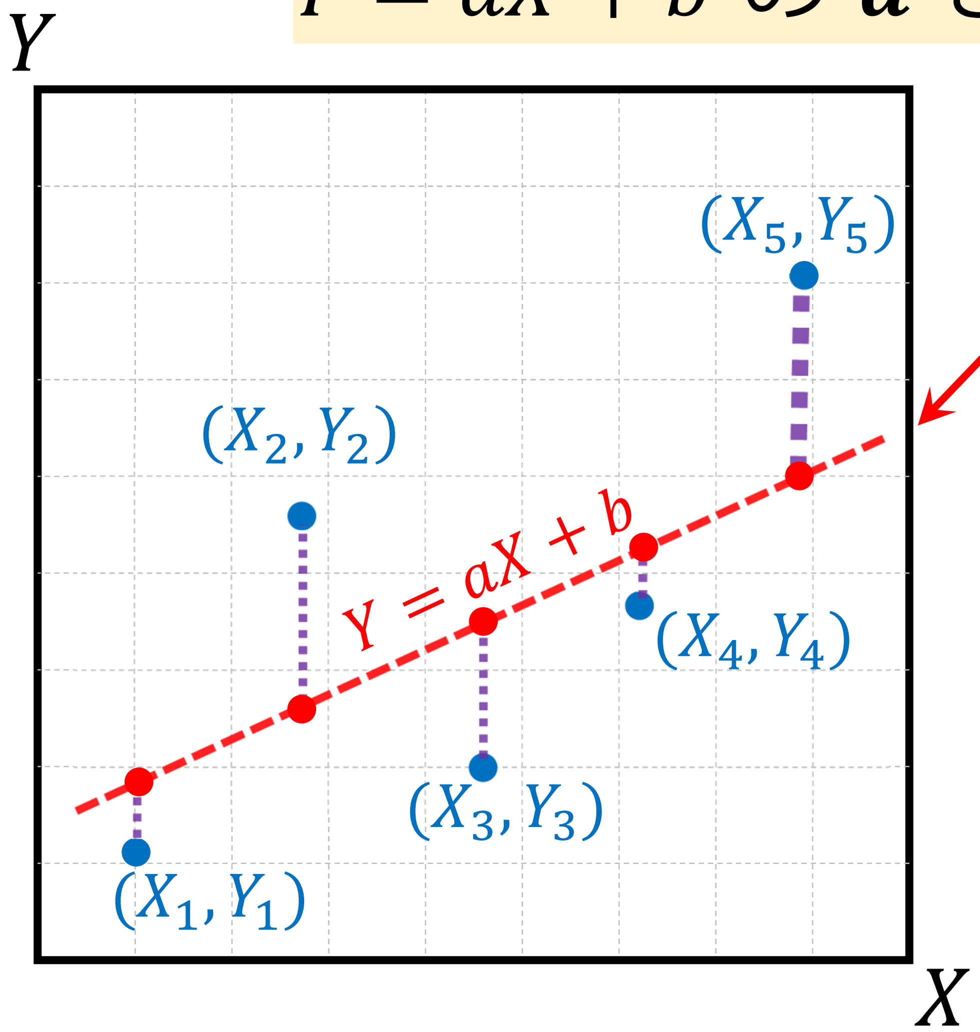
最小二乗法

$Y = aX + b$ の $\boldsymbol{a}$ と $\boldsymbol{b}$ を決めたい。



最小二乗法

$Y = aX + b$ の $\boldsymbol{a}$ と $\boldsymbol{b}$ を決めたい。



それぞれのX_iでの予測値$Y_i{}'$

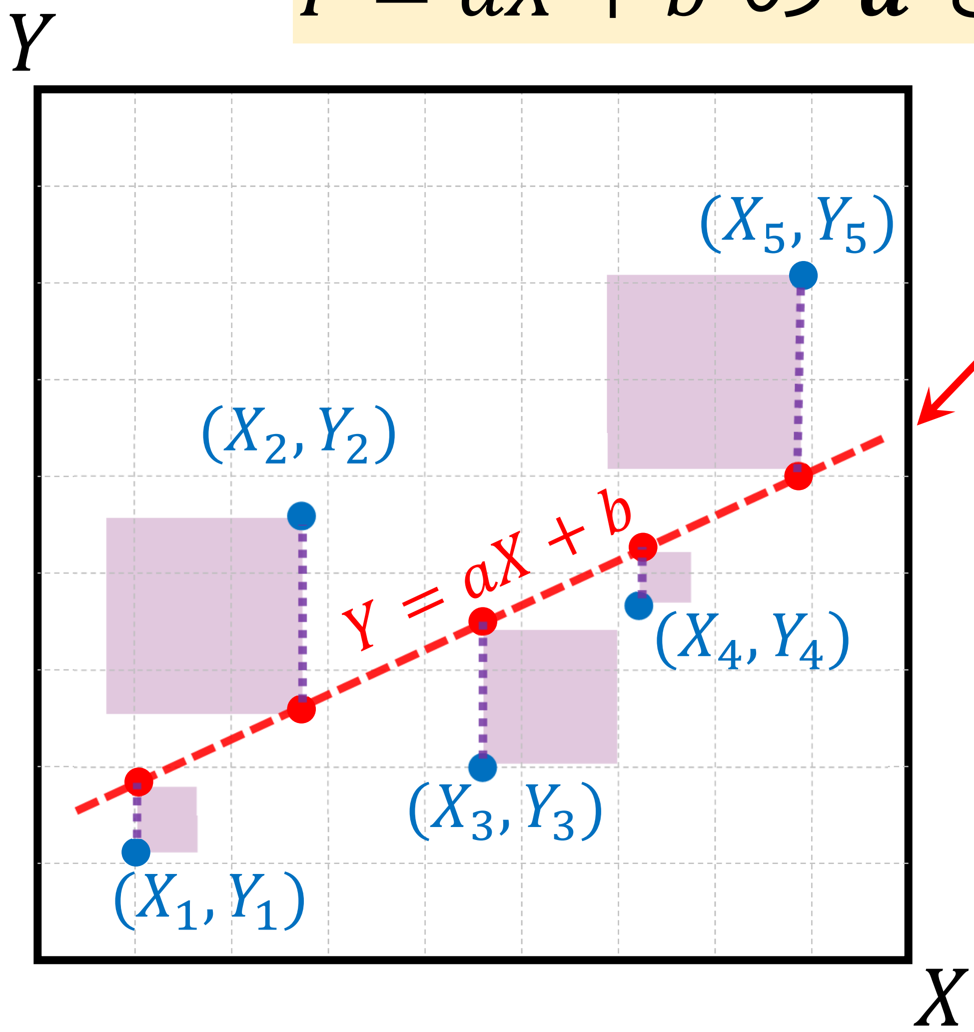
$$Y_i{}' = aX_i + b$$

誤差 $E_5 = Y_5 - Y_5'$

$\qquad = Y_5 - aX_5 - b$

最小二乗法

$Y = aX + b$ の $\boldsymbol{a}$ と $\boldsymbol{b}$ を決めたい。



それぞれのX_iでの予測値$Y_i{}'$

$$Y_i{}' = aX_i + b$$

誤差 $E_5 = Y_5 - Y_5'$

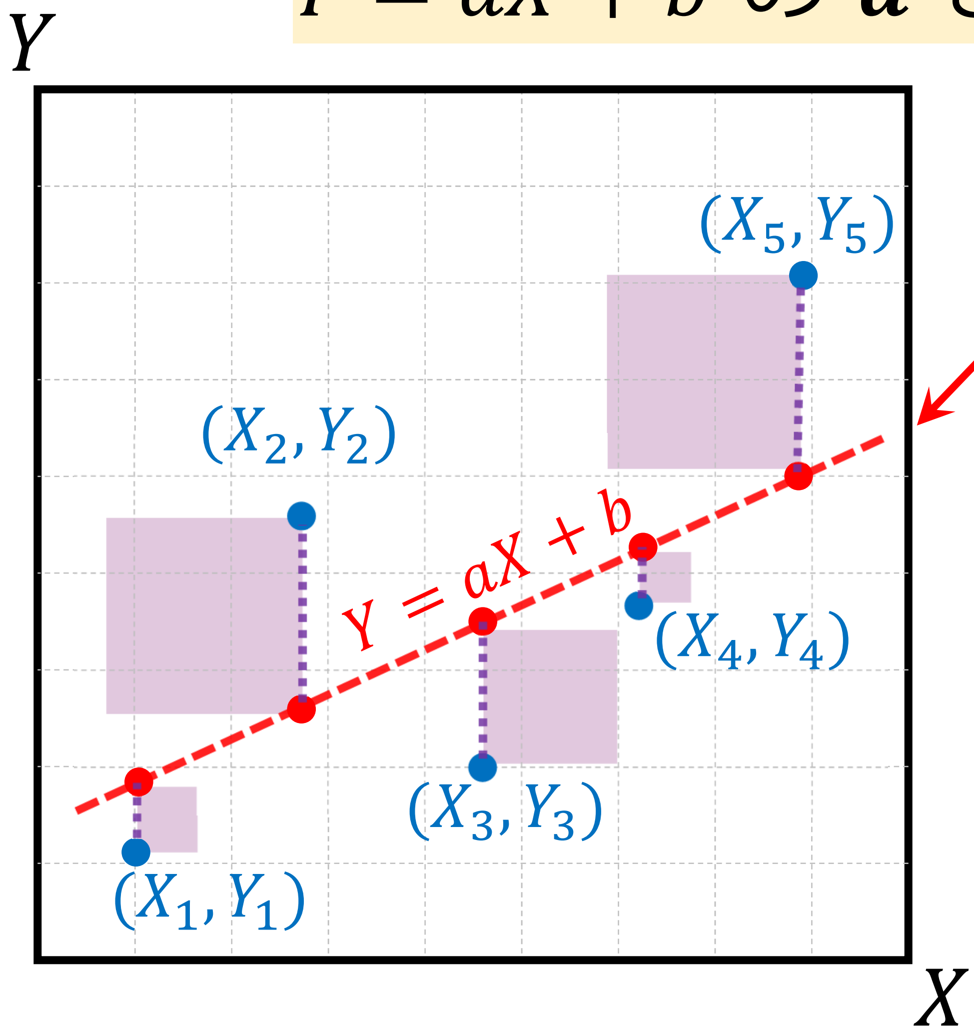
$\quad\quad = Y_5 - aX_5 - b$

■の面積が誤差の2乗

2乗にすることで、
正の値と負の値を
キャンセルできなくする。

最小二乗法

$Y = aX + b$ の $\boldsymbol{a}$ と $\boldsymbol{b}$ を決めたい。



それぞれのX_iでの予測値$Y_i{}'$

$$Y_i{}' = aX_i + b$$

誤差 $E_5 = Y_5 - Y_5'$

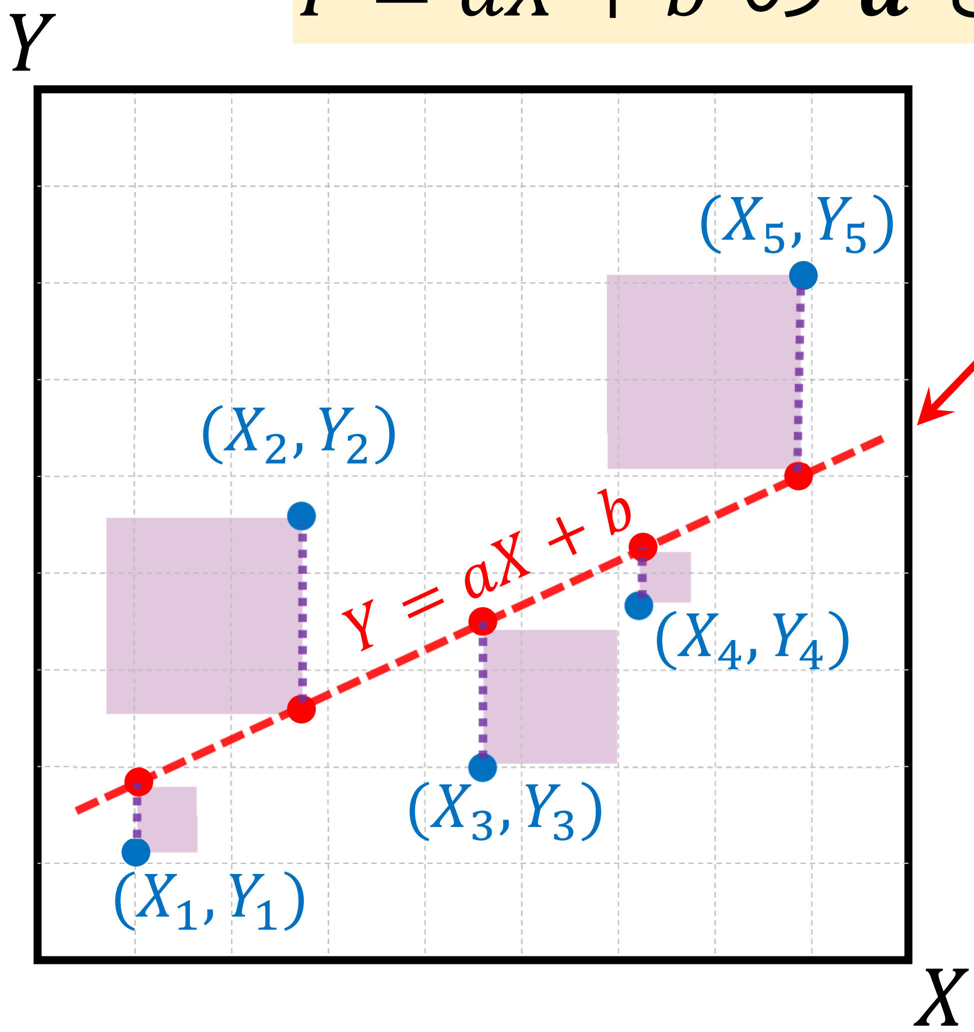
$$= Y_5 - aX_5 - b$$

誤差の2乗の合計

$$E_{\mathrm{total}} = \sum_{i=1}^{N} (Y_i - aX_i - b)^2$$

最小二乗法

$Y = aX + b$ の $\boldsymbol{a}$ と $\boldsymbol{b}$ を決めたい。



$Y_i{}' = aX_i + b$

誤差の二乗の合計とが最小になるようなaとbを求める!!

誤差の2乗の合計

$$E_{\mathrm{total}} = \sum_{i=1}^{N} (Y_i - aX_i - b)^2$$

最小二乗法

導出の計算はこちら
https://manabitimes.jp/math/942

まず直線の傾きを求める

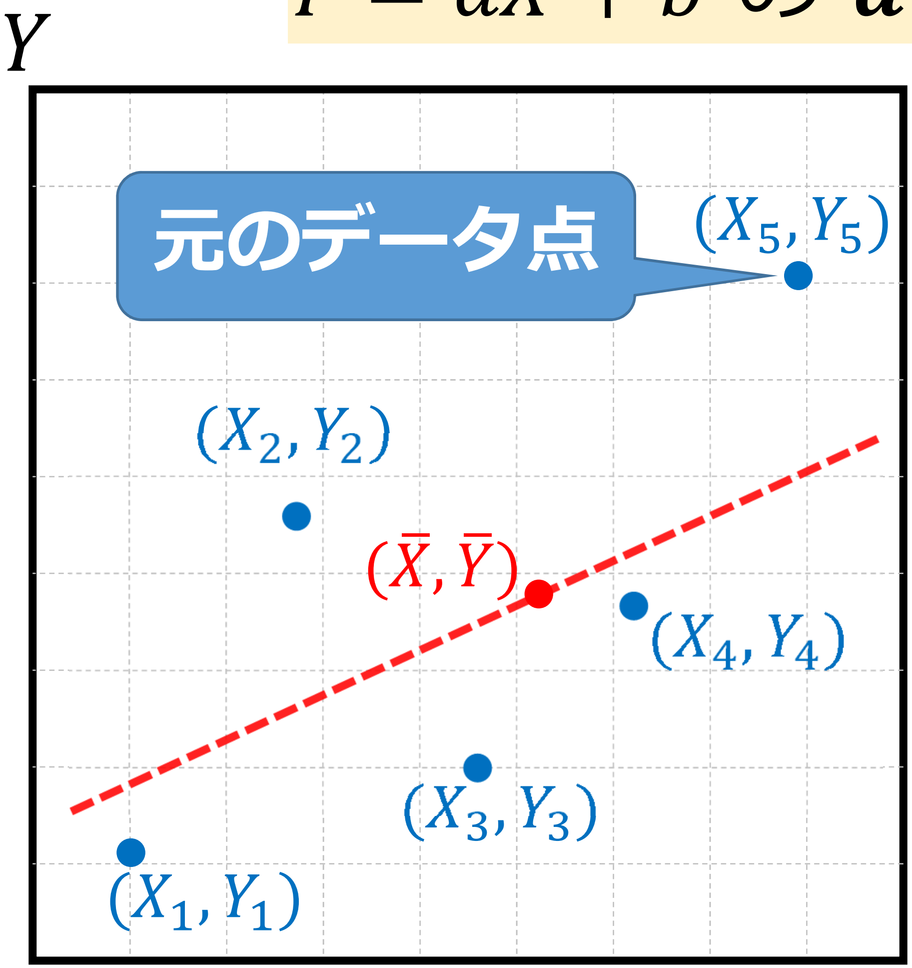
$$a = \frac{s_{xy}}{\sigma^2} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}$$

x と y の共分散 $$s_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

x の分散 $$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})$$

最小二乗法

$Y = aX + b$ の $\boldsymbol{a}$ と $\boldsymbol{b}$ を決めたい。

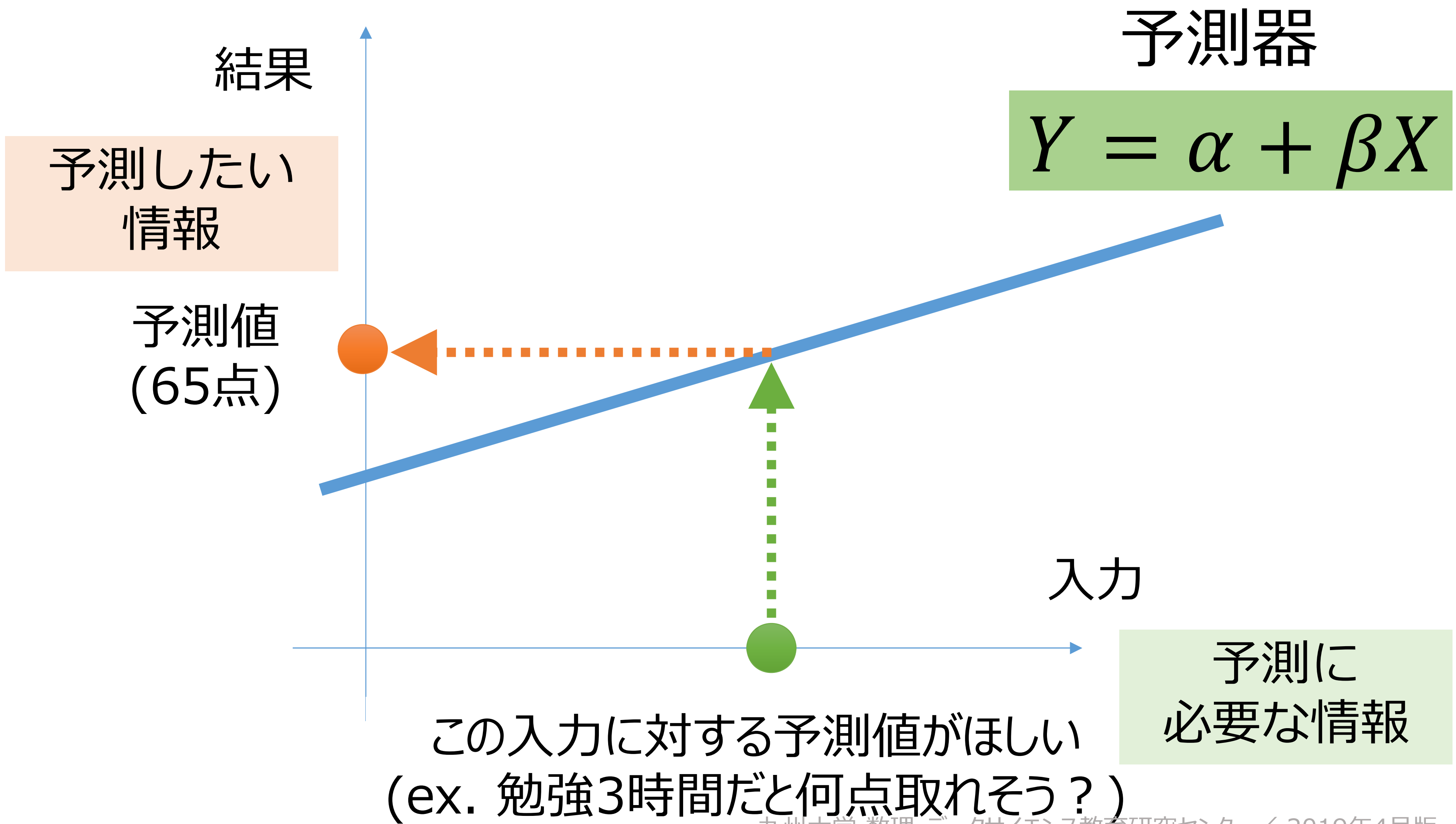


$$a = \frac{s_{xy}}{\sigma^2} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}$$

回帰直線は、
$(\bar{X}, \bar{Y})$ を通るので、

$$b = \bar{Y} - a\bar{X}$$

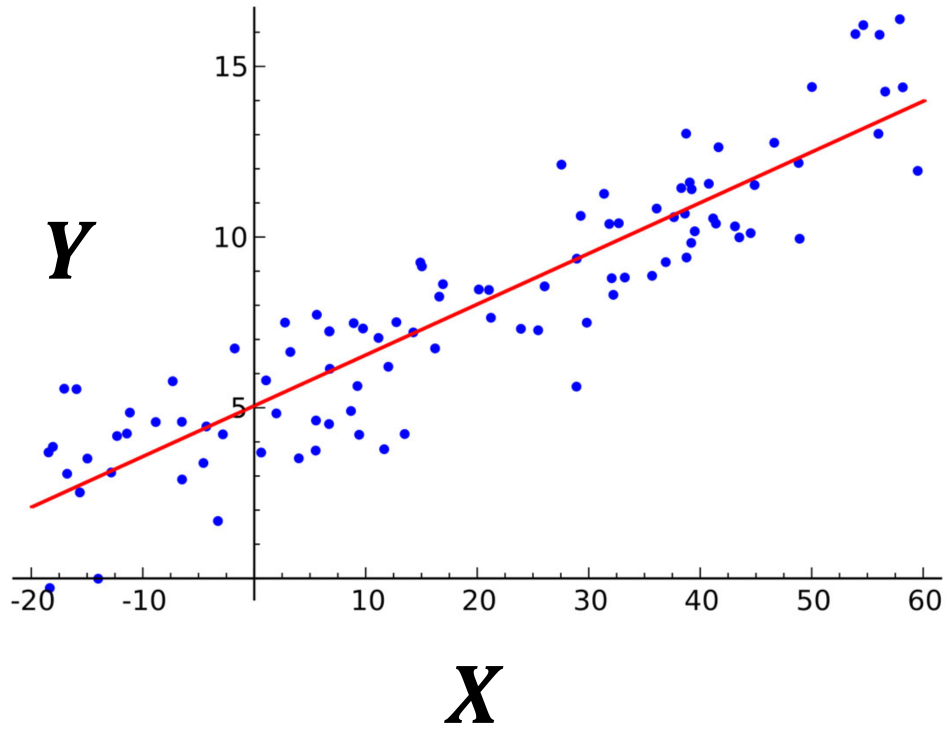
最終目標：予測


予測器
$Y = \alpha + \beta X$
結果
予測したい
情報
予測値
(65点)
入力
予測に
必要な情報
この入力に対する予測値がほしい
(ex. 勉強3時間だと何点取れそう？)

予測の評価

- $\boldsymbol{\alpha}$: **回帰定数**
  $\boldsymbol{\beta}$: **回帰係数**

$$Y = \boldsymbol{\alpha} + \boldsymbol{\beta} X$$

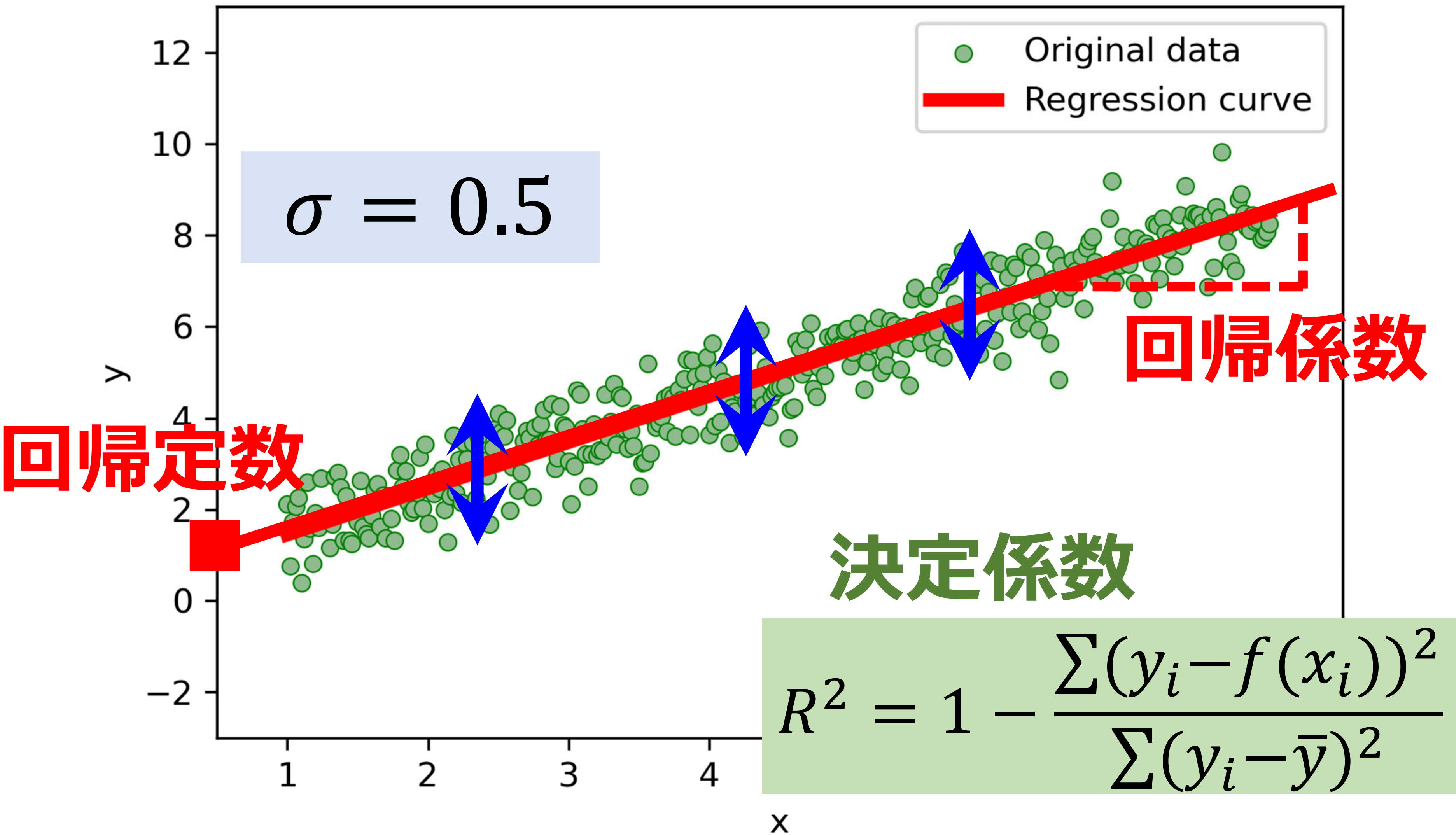


- **決定係数**

モデルがどれだけ
データにフィットしているか
を示す指標

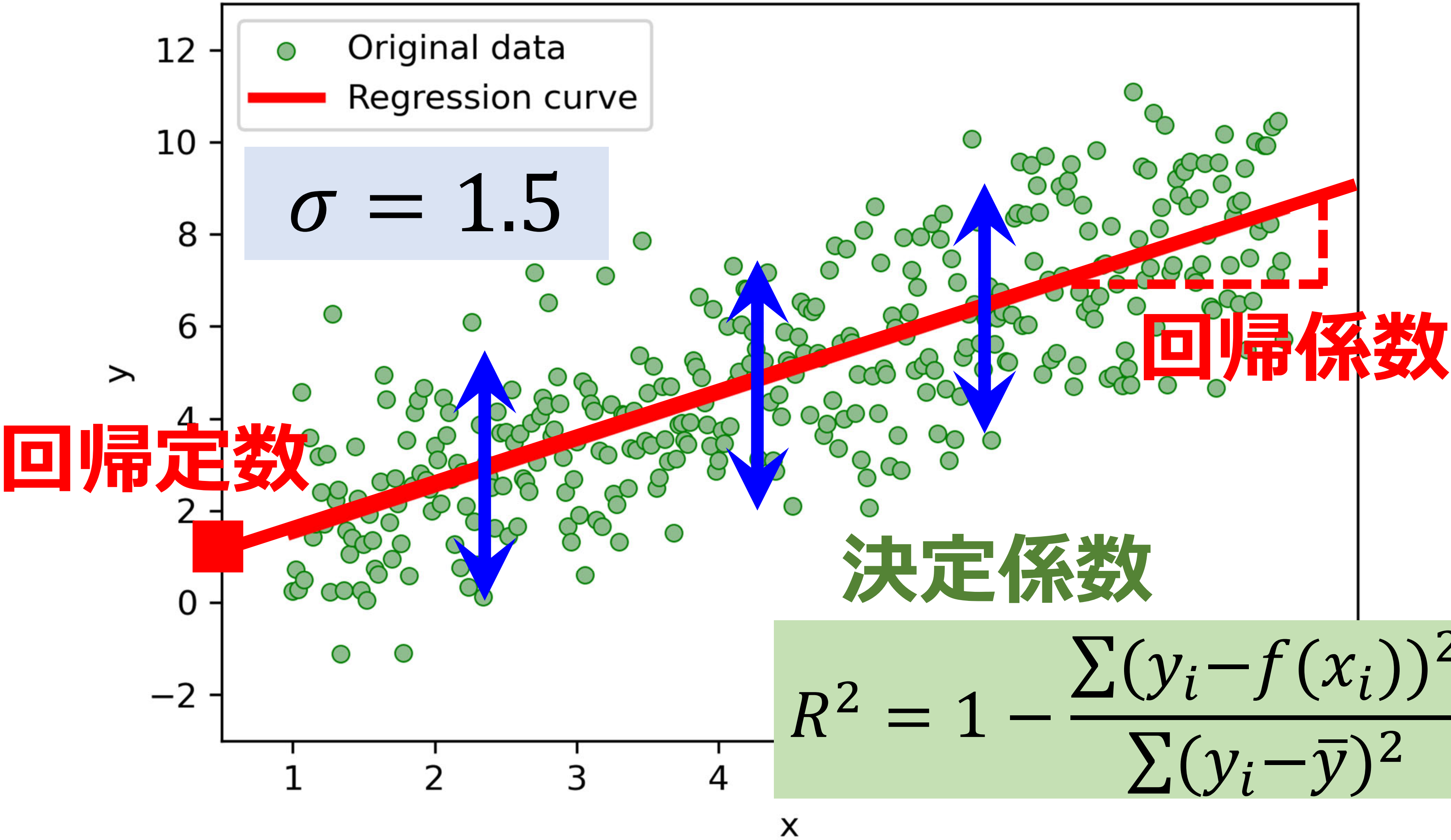
$$R^2 = 1 - \frac{\sum(y_i - f(x_i))^2}{\sum(y_i - \bar{y})^2}$$

モデル予測精度の評価



Original data
Regression curve
$\sigma = 0.5$
回帰係数
回帰定数
決定係数
$R^2 = 1 - \frac{\sum (y_i - f(x_i))^2}{\sum (y_i - \bar{y})^2}$
y
x
12
10
8
6
4
2
0
−2
1
2
3
4

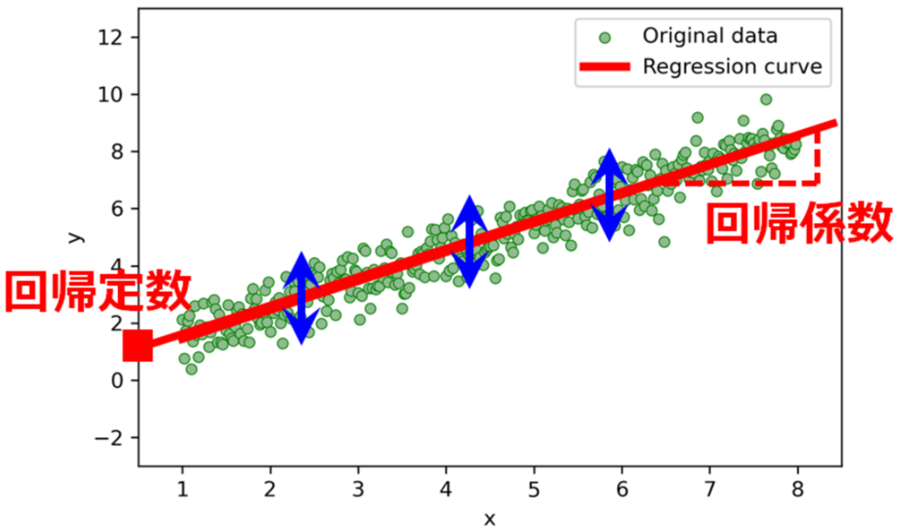
モデル予測精度の評価



$$R^2 = 1 - \frac{\sum(y_i - f(x_i))^2}{\sum(y_i - \bar{y})^2}$$

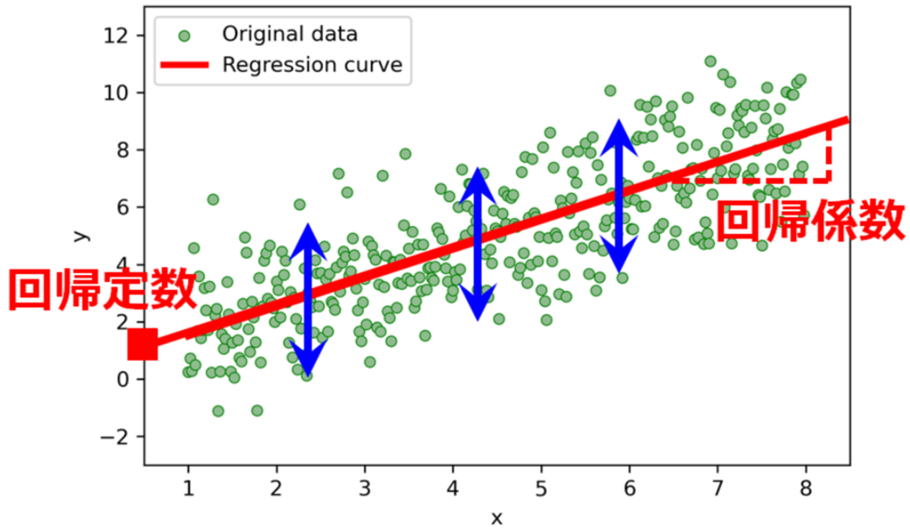
単回帰分析

$\sigma = 0.5$



決定係数0.92

$\sigma = 1.5$

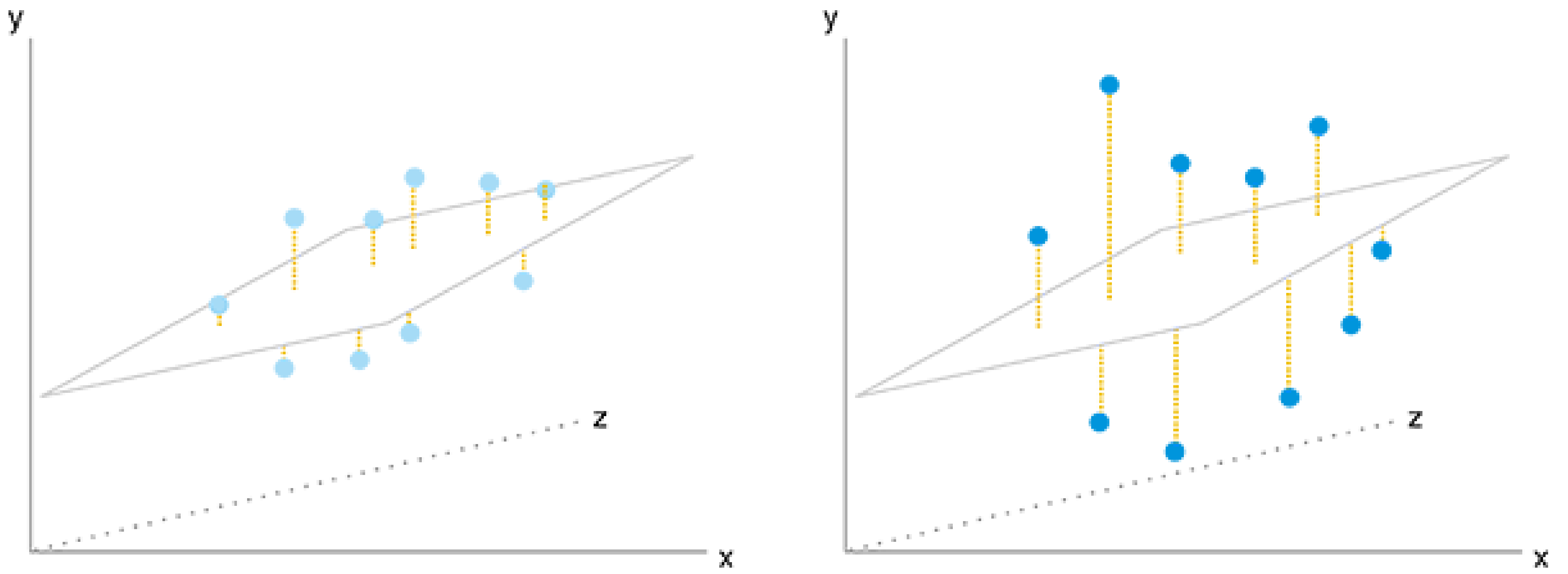


決定係数0.63

$$R^2 = 1 - \frac{\sum(y_i - f(x_i))^2}{\sum(y_i - \bar{y})^2}$$

ばらつきが小さいほうが、
右辺第2項が小さくなるので、
決定係数は大きくなる

重回帰分析



変数の種類が増えるだけで、基本的な原理は一緒!!

https://xica.net/magellan/marketing-idea/stats/about-coefficient-of-determination/

相関係数 (correlation coeff.)

$$R = \frac{s_{xy}}{\sigma_x \sigma_y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

x の標準偏差

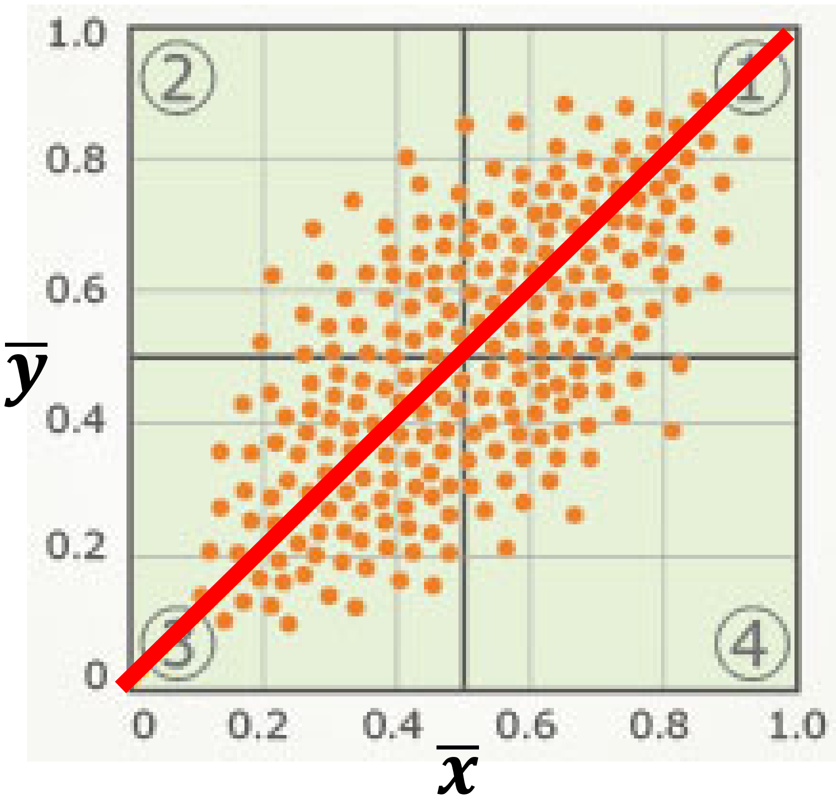
$$\sigma_x = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

y の標準偏差

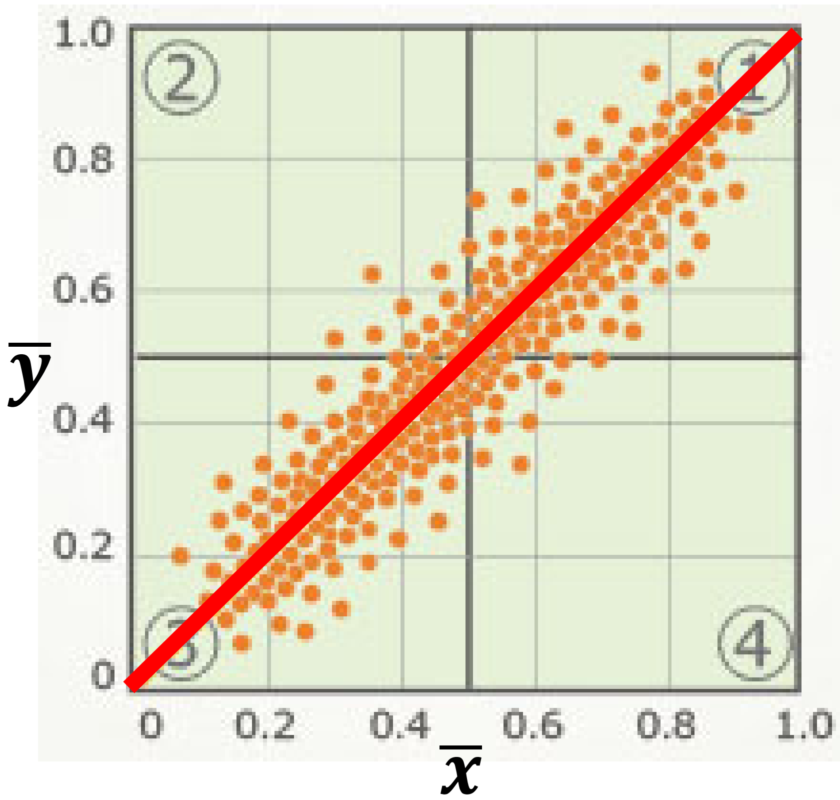
$$\sigma_y = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

相関係数 (correlation coeff.)

(1) 相関が低い



(2) 相関が高い

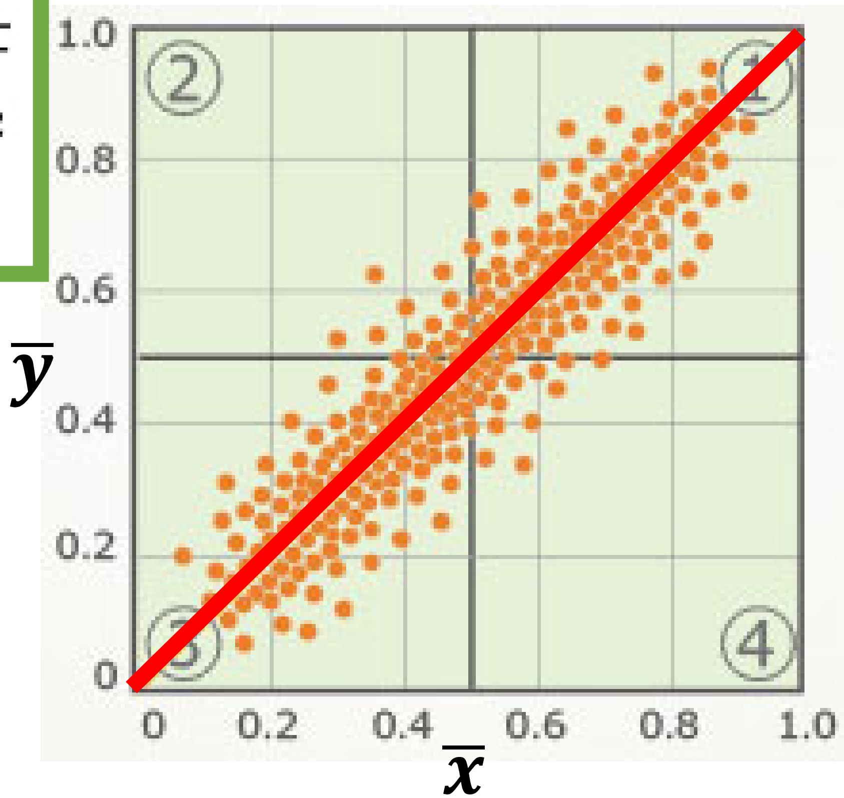


$$R_1 < \mathrm{R}_2$$

相関係数 (correlation coeff.)

$$R = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

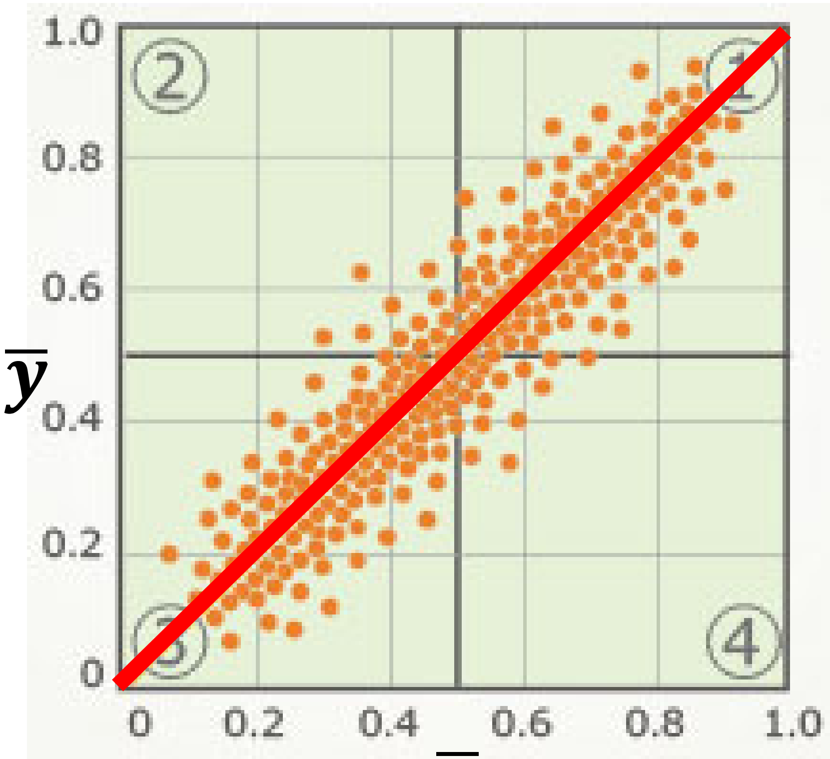
(2) 相関が高い



同じ回帰直線でも、相関が高いほうが予測が当たりやすい。

相関係数 (correlation coeff.)

正の相関

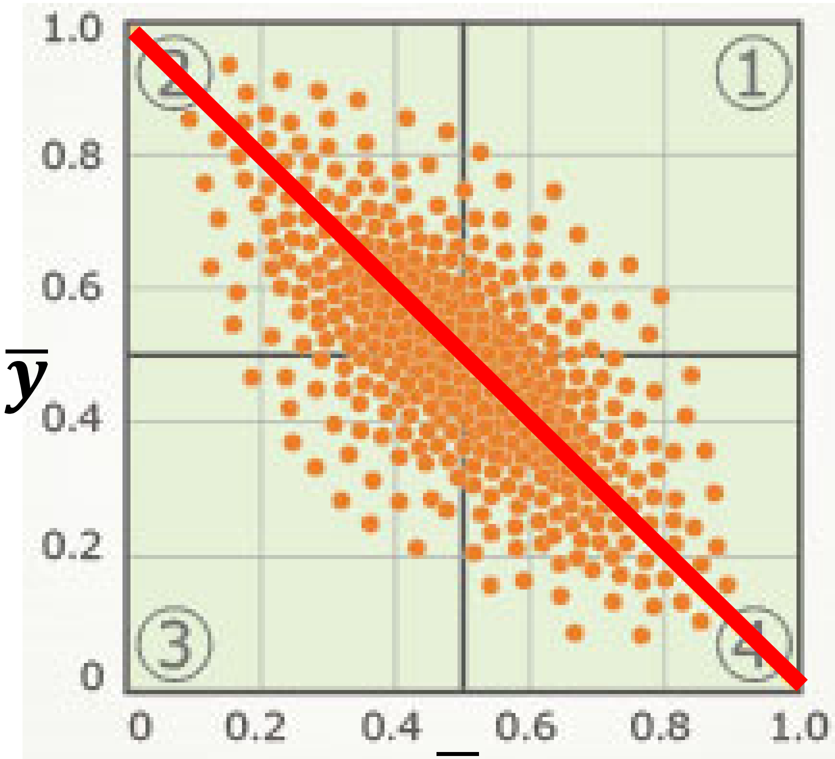


$\overline{y}$

$\overline{x}$

$R > 0$

負の相関



$\overline{y}$

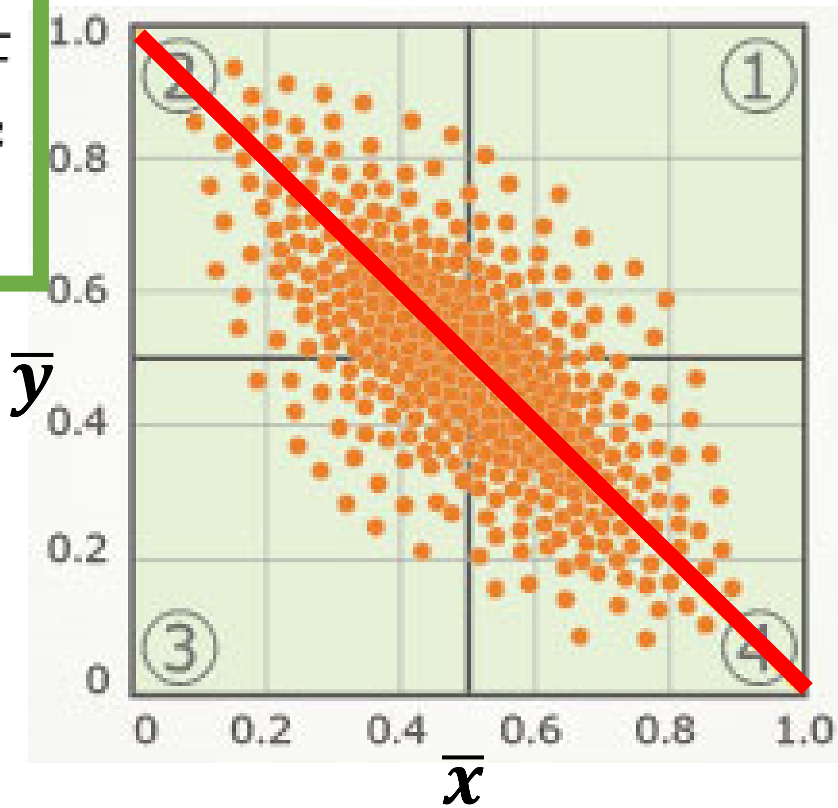
$\overline{x}$

$R < 0$

相関係数 (correlation coeff.)

$$R = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

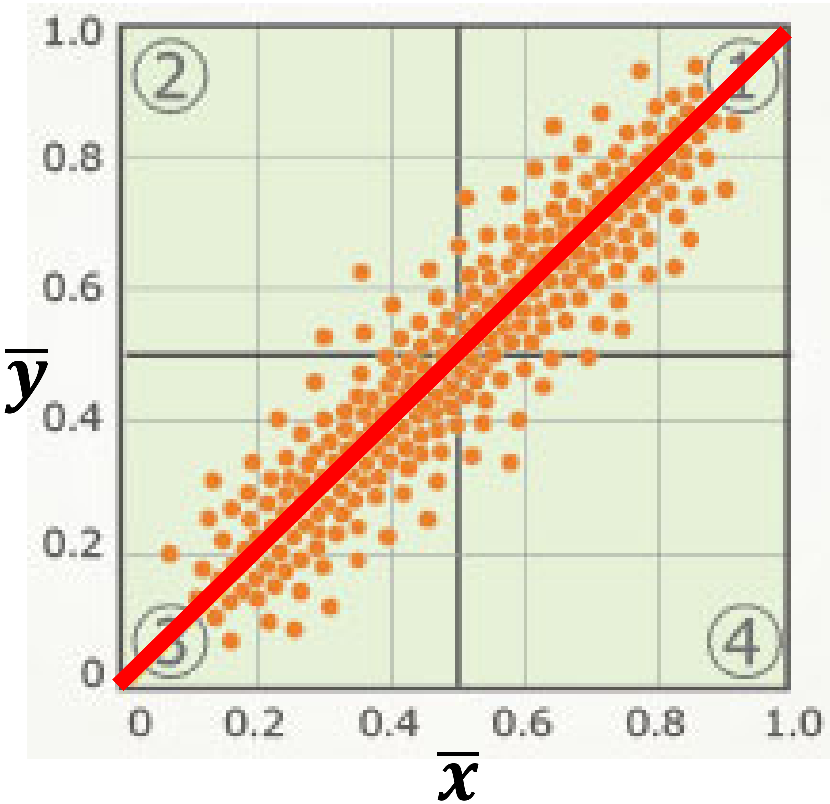
負の相関



$R < 0$

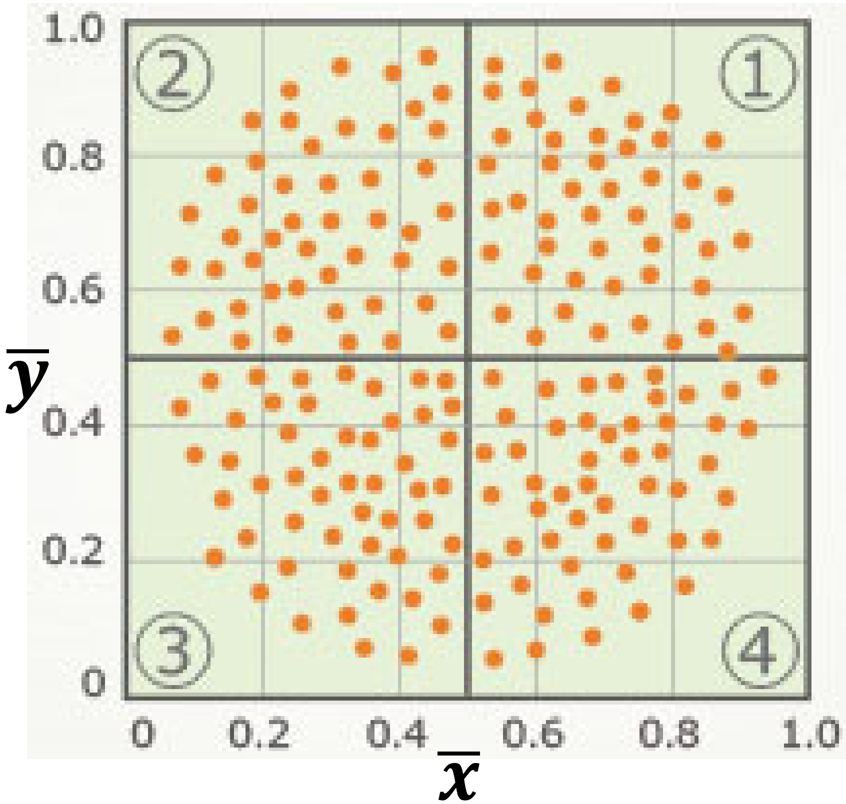
相関係数 (correlation coeff.)

正の相関



$R > 0$

無相関

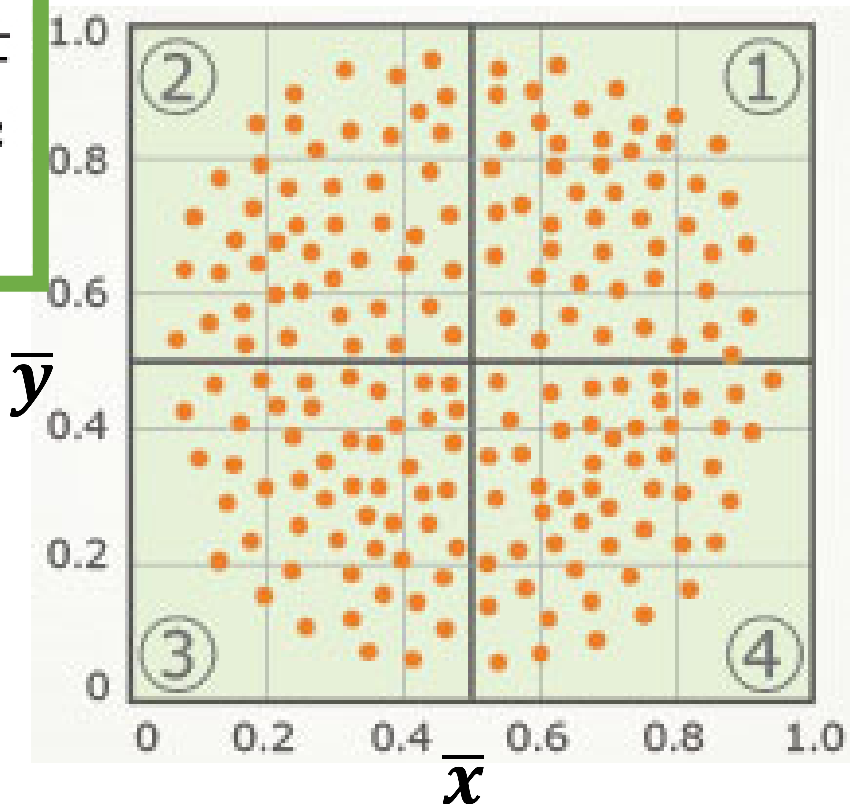


$R \sim 0$

相関係数 (correlation coeff.)

$$R = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

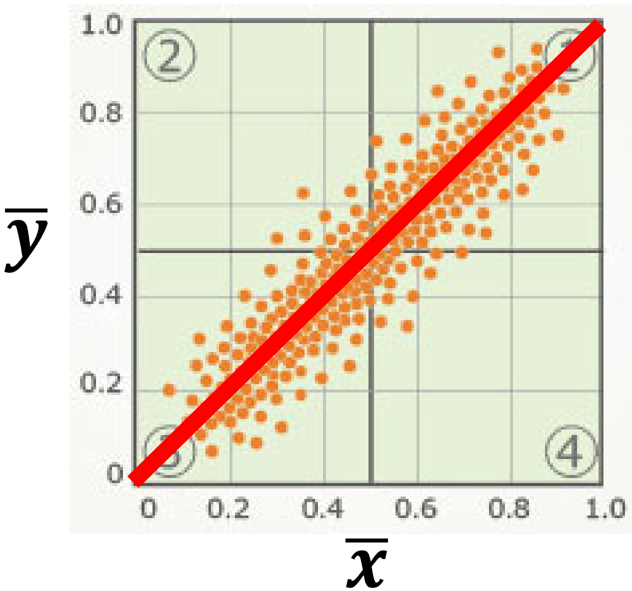
無相関



$R \sim 0$

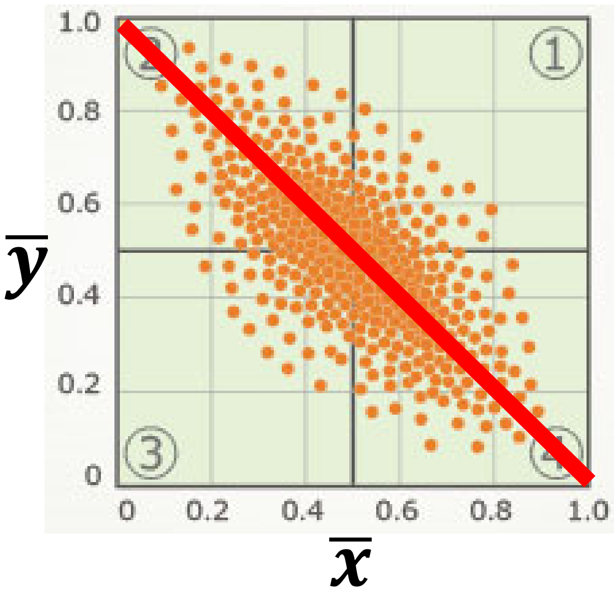
相関係数 (correlation coeff.)

正の相関



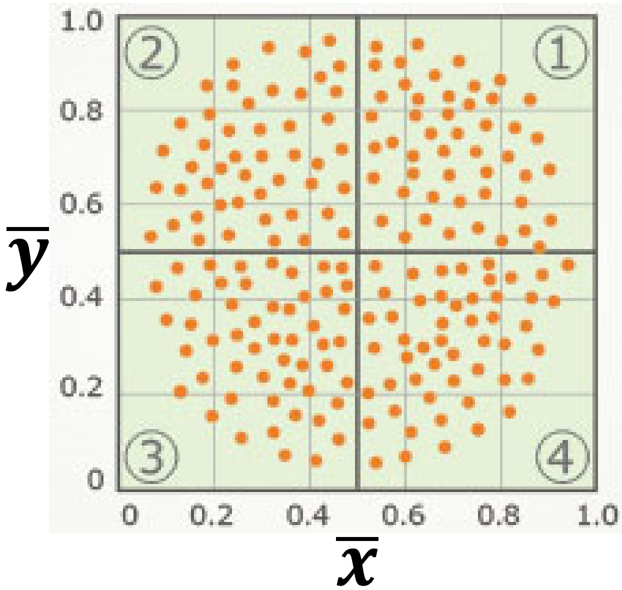
$R > 0$

負の相関



$R < 0$

無相関



$R \sim 0$

相関が強ければ、より良い予測ができる。

Scikit-learnで
回帰分析

いろいろな分析の共通点

5週目：k-means法

6週目：単回帰分析

7週目：ロジスティック回帰分析

見た目全く違う解析ですが、
Scikit-learnを用いた解析という意味では、**手順がほとんど同じ**です。

ですので、k-means法で課題をきちんと解けなかった人は、
6週目、7週目の内容についていくのが難しくなります。

というわけで、手順の詳細をまとめてみました。

解析の手順

1. scikit-learn のインポート
2. データの読み込み
3. headでデータの内容を確認
4. shapeでデータのサイズを確認
5. isnulで欠損値の確認→あれば削除(補完など。。)
6. インスタンスを作成
7. .fitを実行
8. .predictを実行

これはいつもの手順

大事なのはここ!!

(1) scikit-learn のインポート

Numpy, Pandasの時と一緒

```
from sklearn.cluster import KMeans
```

```
from sklearn import linear_model
```

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
```

(2) データの読み込み

乱数を使って
サンプルデータを
生成することもある。

■ CSV ファイルの読み込み

```
df = pd.read_csv('data/w5_rep_lattice.csv')
```

１列目をインデックスに入れるかどうか、オプションで指定できます。

https://note.nkmk.me/python-pandas-read-csv-tsv/

(2) データの読み込み

■ Scikit-learnのサンプルデータの読み込み

アヤメのデータや、乳がんのデータなど、いろいろあります。

```
from sklearn import datasets
iris = datasets.load_iris()
```

このままだとirisはBunch型という変数になります。

```
iris_df = pd.DataFrame(iris.data, columns=iris.feature_names)
```

試しにtype(iris)　と実行してみましょう。
Bunch型のままでも解析できるのですが、
DataFrameで統一しました。

そのため、この行を使って、DataFrameに変換しています。

(3) headでデータの内容を確認

df.head()

で、データの中身を確認。

どんな変数が格納されているか、見てみましょう。

何回もやりましたね!!!

(3) headでデータの内容を確認

df.head()
で、データの中身を確認。
どんな変数が格納されているか、見てみましょう。

何回もやりましたね!!!

(4) shapeでデータのサイズを確認

それぞれの列にいくつNaNがあるか

df.shape
で確認。

(4) shapeでデータのサイズを確認

それぞれの列にいくつNaNがあるか

df.shape

で確認。

何回もやりましたね!!!

(5) 欠損値の確認→あれば削除

それぞれの列にいくつNaNがあるか

df.isnull().sum()

欠損値が一つでもあれば、その行を削除

df = df.dropna(how='any')

本当は、この後、

df.isnull().sum()

をもう一回やって、欠損値が本当になくなったことを確認したほうがいい。

df.shape

をもう一回やって、欠損値削除後のデータサイズも確認したほうがいい。

(5 ’) データの分割

分割したデータを
訓練データと検証データ（7:3）に分割

```
from sklearn.cross_validation import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.3,
random_state = 101)
```

X_train, Y_train: X座標、y座標の訓練データ
X_test, Y_test: X座標、y座標の検証データ

手元にあるデータから、予測モデルを立てた後、モデルの精度を検証するのに使うデータを分けておく。
test_size = 0.3　は、全体の30%を検証用データとする、ということ。

(6)インスタンスを作成

使う関数が違うだけ!!
オプションは関数によって、あったりなかったり。。

```
kmeans = KMeans(init='random',n_clusters=3)
```

```
model = linear_model.LinearRegression()
```

```
logmodel = LogisticRegression()
```

（7）.fitを実行

#k-means法を実行

```
kmeans.fit(X)
```

単回帰分析を実行

```
model.fit(x, y)
```

xが1列だと単回帰分析
xが複数列あれば重回帰分析

ロジスティック回帰分析を実行

```
logmodel.fit(X_train,Y_train)
```

（8）.predictを実行

\# クラスター番号を予測 / Predict the cluster number.

```
y_pred = kmeans.predict(X)
```

\# 回帰直線を求める→説明変数から、目的変数を予測 / Predict the objective var.

```
reg_y = model.predict(x)
```

\# その事象が起こるか起こらないかを予測 / Predict the incidence

```
predictions = logmodel.predict(X_test)
```

(9) 解析後

k-means法

分類したデータを可視化して確認
→クラスタごとに色分けしてプロット

単回帰分析

決定係数から、予測モデルの精度を確認

ロジスティック回帰分析

混合行列から、予測の精度を確認